SPECIAL COLLECTIONS

UNIVERSITY OF ALBERTA LIBRARY

REQUEST FOR DUPLICATION

I wish a photocopy of the thesis by

CLARKE B. HAZLETT (author)

entitled THE STORAGE AND RETRIEVAL OF MULTIPLE CHOICE ITEMS.

The copy is for the sole purpose of private scholarly or scientific study and research. I will not reproduce, sell or distribute the copy I request, and I will not copy any substantial part of it in my own work without permission of the copyright owner. I understand that the Library performs the service of copying at my request, and I assume all copyright responsibility for the item requested.

THE UNIVERSITY OF ALBERTA

THE STORAGE AND RETRIEVAL OF MULTIPLE CHOICE

ITEMS ON COMPUTER

BY

© [illegible]

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF EDUCATION

DEPARTMENT OF EDUCATIONAL PSYCHOLOGY

EDMONTON, ALBERTA

FALL, [illegible]

THE UNIVERSITY OF ALBERTA

THE STORAGE AND RETRIEVAL OF MULTIPLE CHOICE ITEMS ON COMPUTER

BY

© CLARKE B. HAZLETT

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF EDUCATION

DEPARTMENT OF EDUCATIONAL PSYCHOLOGY

EDMONTON, ALBERTA

FALL, 1969

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "The Storage and Retrieval of Multiple Choice Items on Computer" submitted by Clarke B. Hazlett in partial fulfillment of the requirements for the degree of Master of Education.

ABSTRACT

The purpose of this thesis was to design an information storage and retrieval system for multiple choice test items on the IBM 360/67 computer which would accommodate both measurement and document retrievals.

The data management organization involved the setting up of a sequential file with manually coded indexes, and was specifically implemented for the Department of Internal Medicine, though specifications are given to indicate the generality of the design to other educational areas. Provision was made for reducing user error and maximizing useability of output. Documentation specifying hardware requirements, cost of implementation, human requirements, and source listing of all Fortran IV programs used, was also given.

Modification procedures were provided for updating the information bank.

ACKNOWLEDGEMENTS

The writer is indebted to Dr. S. M. Hunka, the supervisor of this thesis, for his direction and contributions. Acknowledgement is made of all the staff in the R. S. McLaughlin Examination and Research Centre for without its help this study would not have been feasible; in particular Drs. B. Hudson, A. McConnell, and D. R. Wilson are to be thanked.

Appreciation is also expressed to the committee members Drs. V. R. Nyberg and D. W. R. Wilson.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Statement and Importance of Problem

Machines called computers were so named because the only significant work given to them was computation. To describe its potentialities, however, one might consider Flanagan's (1966) suggestion for a more appropriate name--information machine. McCarthy (1966) maintains this information machine is becoming the contempory counterpart of the steam engine that brought on the industrial revolution--the computer heralding the information revolution. Grunberger (1966) shows it has the versatility, logical flexibility, and ability to grow that is not matched by anything short of a living organism.

Thus in the last 15 to 20 years an information science has developed as a result of the use of these machines. Cuada (1966) avoids writing a definition of information science, preferring to indicate the areas of study that can be considered as belonging to this field. Taylor (1966) also avoids a formal definition, but does show how information science can be viewed either from an operational or pedagogical point of view, since it ranges through a spectrum from services at one end (for example, libraries) through system design to basic research.

The use of information storage and retrieval systems is a matter of everyday experience for literate people. The public library, correspondence files, accounting systems, directories, dictionaries, and so on are all information storage and retrieval systems. All are comprised of records to which one may address a variety of allowable questions

with a reasonable expectation of retrieving a selection of records in response to each question.

Operationally all such systems employ only three basic processes --the analysis of records, the derivation of new records from old ones, and the physical displacement of records over a distance. These same processes are used in machine retrieval systems. The discussion of the development of the retrieval system for this study will be held within the frame of reference provided by these three basic processes.

Specifically the study concerns itself with the design, implementation, and evaluation of a system for the storage and retrieval of multiple choice test items. Since the Internal Medicine department of the Royal College of Physicians and Surgeons of Canada expressed a desire for the development of such a system the design was applied to their needs. The emphasis of this study, however, is the wider applicability of such systems in general education. A further objective of the study is that the principles underlying the programming techniques for reducing human error and increasing the useability of output, the file organization, search and indexing procedures, and the incorporation of available computer hardware and facilities at this university will provide a basis upon which other educators and administrators can build similar services for retrieval of large masses of data.

The Department of Internal Medicine was one of the first medical specialties in Canada to use multiple choice questions for its annual examination for entrance to the Royal College of Physicians and Surgeons. This department also saw the need for preserving acceptable items for the development of a suitably large bank from which future examinations could be constructed. The problems involved in handling such a bank by doctors

and their clerical staff constitutes an inefficient use of their time and effort. Thus the need was expressed for a machine-based retrieval system.

Spring (1967) found a deficiency in the research literature concerning the application of information science in medicine:

> . . . every article on the subject either outlines the desired scope and then describes one or two subsystems in development, or describes a system so tailored to the situation in which, and for which, it has been developed that it has little applicability elsewhere . . . the literature is devoid of reports of the applicable, broadly useful, complex, debugged, effective, functioning, multipurpose systems so needed in medicine . . . most applications are truly in the research stage . . . . [p. 312]

As shall be seen in chapter two this problem is not the only one encountered when one attempts to find an appropriate system that could be applicable to the need the doctors expressed. It was felt, therefore, that the task of designing and implementing a system for retrieving multiple choice examination items was most worthwhile. Not only would it serve the needs of Internal Medicine, but also other departments--Surgery, Obstetrics and Gynecology, etc. There would of course be as wide a need and application in any educational department using similar examination questions.

The particular advantage of using a computer for retrieval is pointed out by Baruch (1966). He feels that computers greatest assistance is

> . . . in those areas of medicine where the computer can act as an adjunct to the human in tasks that intelligent humans seldom do particularly well. The areas of sorting, filing, indexing, searching, and particularly of being alert for low probability occurences are the kind of 'light thinking' that computers can do well and that intelligent people do poorly [p. 27].

It is reasonable to assume Baruch's conclusion applies not only to medicine, but to other disciplines as well, including education and test development.

Procedure Followed

The design and implementation of this system was developed for use on the IBM 360/67 computer using Fortran IV programming language and syntax. The original bank of items was stored on a formatted tape but during retrieval executions was transferred to the direct access medium of disc through which a sequential search was made. However, since the system design is an integral part of this study it will be dealt with in detail in Chapter III.

Limitations of the Study

Information and retrieval systems operate as follows. Records are gathered and inserted into a collection, possibly with indexing to give some orderly manner. The user addresses a question to the collection and on this basis a search is made with pertinent records being retrieved. It is important to note that records are created and organized before the specific questions a system is to answer have been stated. That is to say, the system is created in anticipation of needs that are not fully known.

A meaningful question a designer must ask then is, "Is it possible to devise an information storage and retrieval system that will conveniently retrieve pertinent records in response to all possible questions?" Unfortunately it is not. According to Lipetz (1966),

> Not only is it impossible to create an information storage and retrieval system that will respond to all possible questions but also it would be prohibitively expensive to try to approximate such a condition. In practice all information systems and retrieval systems must adopt a more modest object [p. 178].

Designing systems to satisfy unstated needs may sound impossible, yet systems are being designed to do just this. By extrapolating from past interests and trends, systems can be developed that give somewhat adequate retrievals. Indeed, this probably is the only rational approach to design analysis.

With this in mind the following limitations were imposed on this study. (*a*) A sequential, and not a random or list file organization, (the explanations of which will be found in Chapter II) was used. (*b*) No attempt was made to program the computer to do automatic indexing; only manual indexing was incorporated. (*c*) A simple four level hierarchy determined the ranking of relevant output. (*d*) One iterative procedure was provided to improve user satisfaction.

Since each of these limitations is an integral part of the system design (Chapter III) and because the decisions made for their incorporation are made on the basis of the suggestions of other researchers (Chapter II) they are dealt with in detail in these later chapters.

Definition of Terms

There are no specific connotations of any terms used in this study which are foreign to the field of information science. However, because this study may hold some interest for educators who do not necessarily have a background in the computing field, some of the more basic terms in the vernacular of information science are presented which are not defined later in this text:

block--the number of records brought into the active memory of the computer;

core--the active memory and workspace in a computer;

data set--the media containing data such as cards, tapes, disk, etc.;

disk--a direct access medium used by computer in lieu of tape, cards, etc.;

formatted--information specified to the computer in non-binary characters;

logical record length--number of characters or variables in a record;

record--that which is defined by one READ or WRITE statement;

unformatted--information specified without the control of the programmer and contained in binary characters;

information retrieval systems--systems which retrieve facts (measurements) in response to requests;

reference retrieval system--system which retrieves documents or citations in response to requests.

There are a number of other terms with which the reader may be unacquainted. An attempt is made, however, to explain those terms as the system design unfolds in later chapters.

CHAPTER II

SOME PERTINENT LITERATURE

Introduction

The problems involved in assessing library automation literature are considerable. Some are due to the complexity of the library processes and the computer techniques being described. Others are due to the manner in which library automation literature is being produced and published, or for that matter--not published. A proliferation of formal documentation exists in addition to private channels of circulation. Markuson (1967, p. 255) maintains the reason for such communication channels is due to the fact that only relatively few people are engaged in library automation activites and that they are beset by continuous, often conflicting, demands on their time. Thus when a retrieval system begins to function, any documentation that results is often through informal channels of communication. A reviewer must be concerned, therefore, with tracking down elusive items if he is to be up-to-date on current work. The use of Cuadra's three volumes of *Annual Review of Information Science and Technology* (1966, 1967, 1968) and the Association for Computing Machinery publications, *Computing Reviews*, Volumes 7-10 (1966, 1967, 1968, 1969) are of particular help to a reviewer in finding relevant studies, and this reviewer acknowledges their contribution to this chapter.

Besides the difficulty of finding relevant articles on library automation Black and Farley (1966) point out that:

> almost none of them include system specifications, design specifications, design projections, personnel requirements or procedural manuals, or such important details as input card formats, tape formats, outputs, running time, costs, problems encountered, or solution thereto [p. 386].

Thus the lack of writing in this field at a high level of practical detail and concentration upon the use of equipment available to the institutions in which the work has taken place makes it difficult to assess the relative worth of the studies. Hypothetical installations still on the drawingboard may offer promise for the future but provide little help to the present attempts to design information systems.

The format of the remainder of this chapter will be as follows. First, a discussion of the principles and practices underlying the theory of information retrieval will be provided. Following this, a summary will be given of the literature related to such design specifications as (*a*) file organization including sequential, random, and list files, (*b*) data coding techniques with particular reference to manual and automatic indexing and (*c*) associative techniques in file searching. Finally a review will be made of some of the more noteworthy retrieval systems already in operation in this field. As such they provide a backdrop against which an evaluation may be made of their relevance with respect to the particular design involved in this study.

Principles and Practices Related to Information Theory

Vickery (1965) maintains ". . . there is yet no unified theory of retrieval systems and a good deal of retrieval practice is still an empirical art, unsullied by theory [p. 399]." Saltzbery (1963), however, tries to delimit the area by dealing with three underlying principles of

the quantitative aspects of the efficient storage and communication of information--the measure of information, the storage of information, and the communication of information.

Information Measure

Although information theory is essentially a mathematical subject, a basic understanding of the underlying principles can be acquired without resorting to complex mathematical arguments.

Saltzbery (1963) defines one bit of information as the amount of information necessary to resolve two equally likely alternatives. If the uncertainty is greater, then the amount of information necessary to remove it is greater. It follows that a message which identifies one of eight equally likely alternatives contains more information than a message which resolves only four equally likely alternatives.

Consider for example, a simple game in which one is asked to guess a number between one and eight. With no *a priori* knowledge, the probability of correctly guessing is 1/8. In the language of information theory one would ask: how much information is needed to resolve the receiver's uncertainty assuming one only asks questions to which the answers can only be yes or no? In this example the minimum number of such questions is three. Is the number greater than four? If the answer is yes, is the number greater than six? If the answer is no, is the number five? If the answer is no one has resolved all his uncertainty: the number must be six. Hence the receipt of one bit of information reduced the probability from 1/8 to 1/4, two and three bits of information reduced the probability to 1/2 and 1 respectively.

This illustration serves to introduce one to the basic concepts from which the quantitative definition of information can be obtained, namely, the number of equally probable states of a system (here eight), the number of alternatives resolved by each question (here two, because of the binary nature of the question) and the minimum number of questions necessary to determine the state of the system (in this case, three). Thus the relationship is defined as $2^3=8$. In the vernacular of information theory one would say that three bits of information are necessary to determine the state of such a system. That is, three appropriately chosen questions, each of which resolves two alternatives, reduces indeterminancy to certainty; the problem of choosing appropriate questions is analogous to that of choosing an appropriate code which for this study will be dealt with in Chapter III (cf. pp. 41-46).

Information Storage

The problem of storing information is essentially one of making a representation. This representation can take any form as long as the original can be reconstructed at will. Therefore, one simply has to ensure that every possible event is recorded and can be represented in the information bank. The capacity of the bank simply is the total number of listing states which it will admit. This topic will be dealt with under blocking factors in Chapter III (cf. pp. 74-76).

Communication of Information

Now let us consider information theory as it pertains to the communication of information. If one assumes information received is the difference between the state of knowledge of the recipients before and

after the communication, then one can understand Saltsberg's (1963, p. 11) more precise definition.

$I = \log \frac{Pa}{Pb}$, where I is information received, Pa is the probability of an event at the receiver after a message is received, and Pb is the probability of an event at the receiver before the message was received. For example, in receiving a message regarding the sex of a baby, where the receiver does not know its sex, Pb=1/2 and Pa=1. Therefore, $I = \log 1/_{1/2} = \log 2 = 1$, that is, 1 bit of information.

However, as Gove (1957, p. 7) points out, all that is being measured in such a case is the number of binary questions (one's and zero's in a computers language) and not the amount of understanding. Hence, while information can be measured, its unit being the bit, the question of whether or not meaning is measureable so far remains unanswered for want of an acceptable unit of measurement. However, Abramson (1963) maintains the semantic aspects of communication are irrelevant to the design problems of efficiently storing, retrieving and sending information.

One further aspect of information theory should be considered, that of information language, which is not a language for programming information retrieval systems, nor is it a query language for interrogation of the system. It is, however, a language used to represent the content of a document. Soergel (1967) enumerates the requirements to be met by such a language: (a) unambiguous, (b) flexible, (c) in natural language or transformable to it, (d) adaptable to the subject matter, (e) capable of detecting incompleteness or inconsistency, (f) capable of encompassing the scope and variety encountered (g) meeting the requirements of

automatic processing equipment, and (h) capable of minimizing the time, cost, and effort requirements of processing.

Information theory thus provides us with certain principles for analyzing and improving the design of storage and communication processes. These principles will be referred to in the design of this system (Chapter III) and its evaluation (Chapter IV).

File Organization

The literature (Dodd, 1969; Salton, 1968, pp. 243-252) gives two main reasons for the utilization of distinctive data files. Firstly, the demands of users differ. Each type of demand may call for different types of file organization, for example, whether or not rapid storage is necessary, how thorough a retrieval is required, time limits imposed on updating the file, need for holding overhead cost at a minimum, nature of output, and so forth. All these factors influence the type of file the user will choose.

Secondly, the characteristics of the medium in which the data is stored need not necessarily be compatible with the demands of the application. Hence, different file organizations help to compromise between user requirements and the physical limitations of the user's computer hardware, such as amount of core storage, number of data sets available, capability of handling processes, and so forth.

Basically all data organizations are built on three types of files --sequential, random, and list. Since file organization forms the heart of an information storage and retrieval system, this reviewer has felt it necessary to review in some detail how these three files work. The

reader is referred to the manual produced by IBM (1966) on direct access storage devices and Dodd's article (1969) on "Elements of Data Management System" for a more thorough and more technical explanation. It is imperative for the reader to understand the data organization methods available in order to understand many of the aspects of the current problem as discussed in Chapter III. Furthermore, such an understanding provides a basis upon which one may evaluate the system used in this study, a discussion of which is provided in Chapter IV.

Sequential Organization

This method, which is also referred to in the literature as Direct File Organization, is undoubtedly the best known. Records are stored in positions relative to other records according to a specified sequence. This sequence may be in order by document number or other common attributes, or records may simply be in the order of their arrival in the bank. In any case items to be retrieved are identified by a sequential scan of the complete file. The ramifications of using such a file are varied--some advantageous, others imposing restrictions.

Salton (1968) states:

> . . . if the information is to be retrievable according to a variety of different keys--for example, subject identifiers, year of publication, author name, publication, and so on--the direct file system is often the only practical one, since it is not usually possible to store many copies of the same file to account for the various desired file orders [p. 244].

The importance of this point made by Salton has very direct implications for this study and the reader is urged to keep it in mind in the reading of subsequent chapters. The response time for sequential file searches is necessarily slow, however, since a complete file scan is generally

needed before any information can be retrieved. In a sequential file search the first key is examined; if the key is not correct the next record is examined and so on, until the correct record is found.

Updating files in direct files is also disadvantageous. If a new record is shorter or longer than the original record, parts or all of the adjacent records would likely be destroyed when the record is rewritten. Even more inconvenience is encountered in updating blocked records. To do so is impossible unless the entire block is rewritten.

It is obvious, therefore, that the rewriting of sequential files is usually done by copying records from one data set to another as needed. This is necessarily expensive and therefore users would only resort to this procedure when a number of records are to be altered.

Another difficulty is encountered if one attempts to insert new, or remove old records from such a file. An insertion requires that already stored records be "pushed apart" and of course the converse is true for the removal of old records. New records of course could be added out of sequence to the end of a file and sorted later into proper sequence. Such a process again leads the user to copying the entire file onto a new data set, such as a new tape.

The difficulties encountered in the above data organization no doubt led system analysts to develop a file which could eliminate some of these difficulties. Such a file was called random organization.

Random Organization

In this system, records are stored and referenced on the basis of the relationship between the key of a record and the direct address of the location where the record is stored. This latter address is used

when a record is stored and used again when the record is to be retrieved. Three methods are generally used for accessing records--direct address, dictionary look up, and calculation.

Direct address is used if the programmer, knowing the precise size and number of records in his data file, is able to supply the direct address at storage and retrieval times. This address is then used to access a record on storage media.

With the method of dictionary look-up a record's direct address is obtained prior to storage or retrieval with both the record's key and its direct address being stored in a dictionary. When a record is stored or retrieved the key is found in this dictionary and the corresponding direct address is used. For example, if the key, "RENAL," is compared with the dictionary and the direct address 1557 is found, then the direct address 1557 is used to store and subsequently, if necessary, to retrieve the record whose key is "RENAL."

The use of a dictionary insures that each record has a unique address. However, to achieve this the dictionary must be large enough to include all potential direct addresses and it may occupy as much space as the data itself. Also the step-by-step sequential search of a dictionary may offset the advantages gained by unique record addresses.

The third method--calculation--involves converting the key of a record to a direct address. This procedure, however, does not necessarily insure that the address is unique. For example, each letter of the key "RENAL" could be replaced by a number--R by 18 (because R is the 18th letter in the alphabet), E by 5 and so on to L by 12. The sum of these numbers is now the direct address of the record whose key is "RENAL." However, the same direct address would be obtained for the key "LANER." This "overflow" is usually handled by pointers; if the retrieved record

with first address is not the desired record, then the pointer with that key is used to retrieve another record having the same calculated address. This sequence is continued until the correct record is found. This principle is used in the IBM 1500 system for retrieving course material.

Bleier and Vorhaus (1968) found the use of this type of file organization had the following advantages. Queries are retrieved rapidly since one could operate upon the list of addresses of records to access relevant records. Since only a small portion of the complete file is examined there is certainly more efficiency in terms of time in this system as compared to sequential files in which the complete file is searched. These researchers also found that the size of a data base has very little effect on the speed of retrieval.

However, Bleier and Vorhaus also point out the disadvantages encountered in a random organization. Firstly, there are increased storage requirements to handle the list of addresses in core. Therefore, one must weigh the cost of core as compared to the cost of time to determine optimal economy of usage. Furthermore, they found a significant increase in the complexity of maintaining the system. This is not a surprising conclusion since one can see that the programming ramifications would be much greater than in the case of sequential files if one had to program the handling of overflow problems or the manipulation of large unwieldly dictionaries. Dodd (1969) points to another disadvantage.

> Although random organization does allow for rapid access of a particular record with a known key, it is not suited for rapidly accessing a number of records. This limitation is imposed by the time taken by the hardware access mechanism to locate a record [p. 122].

One further notation that Dodd (1969) and the IBM (1966) manual point out that is pertinent to this study is that all records used under random organization are generally of a uniform length. The implication of these limitations will be discussed in the first section of Chapter III (cf. pp. 31-36).

List Organization

The use of pointers in the calculation method of random file organization leads one to the third main file organization, that of list files. In sequential organizations the next logical record is the next physical record. In list organizations, however, a pointer accompanying each record serves as an address to the next logical record which may or may not be the next physical record. There are three basic types of list files--the simple list structure, the inverted list structure, and the ring structure.

Simple list structure is in practice a sequential file that may, for example, have physical records at locations 23, 59, 117, 1105 but which may be sequentially, that is logically, retrieved together by the use of pointers. Initially there is a pointer to the first record number 23, that record points to number 59, and so forth. If a record is to be updated, records can be placed anywhere within the list by changing the pointer of the record preceding the new record and inserting a new pointer in the record inserted. Conversely, the removal of an item requires changing the pointer in the preceeding record to point to the record following the one deleted.

There are, however, serious limitations to this type of data management. Firstly, it is a rare retrieval system in which items belong exclusively to one category and have no relationship to other categories. Indeed, as shall later be seen in the section of this chapter dealing with operating systems, most users demand, and systems incorporate, means whereby items can be ranked according to their degree of association with not one but a number of categories. To use this facility in the simple list structure requires that each record have more than one, possibly many, pointers to and from it. Each record must therefore become a member of many lists. A deletion of a record which is a member of several lists then becomes immensely more difficult since one must find the preceding records of all lists that point to the record being deleted. Furthermore, extra pointers must be stored so that the preceding as well as the next record of the list may be found. Similar complexities, of course, are encountered for any additions of records.

Inverted list structure is one in which the restriction on list length is taken to its ultimate conclusion, that is, the list length is restricted to one, and each key appears in the index. The index thus points directly to the record requested and no further pointers are used. Hence, the list has become inverted, a condition which lends to the name applied to this procedure. The reader will see that, in practice, such a file is much the same as a random organization using the dictionary look-up method. Like that procedure, inverted files have the similar advantages, that is of providing relatively efficient access to all data and of being suited to retrieval requests which are less predictable than specific. Similarly the two procedures share the same disadvantages,

the main one being the requirement of a large dictionary to be stored in core.

Ring structure is an extension of list organization, but instead of the last record in a list being given no further pointers, a pointer is made back to the first record. This first record in a particular list is given a special symbol to indicate that it is the first record in that ring. Furthermore, all records pertaining to a given ring list except this first record have two pointers--one to the previous logical record and one to the first record. It will be recalled that records may be included in more than one list structure. The same facility can be used here also.

Such ring structures may prove to be very powerful as they provide a facility to retrieve and process all records in any one ring while being able to branch off at any or each of the records to be retrieved and process other records which are logically related. Such other records would also be stored in a ring structure and in turn permit the same facility; this nesting is carried through to any level required by the logical relationships. However, at any given record in any given list there are two pointers, one being to the previous logical record, the other to the first record of that ring. Therefore, having obtained one record at a given hierarchial level one would also be able to retrieve as many records, with the same hierarchial characteristics, as were in that given list.

To illustrate the use of this procedure, assume a record is retrieved by the key "Allergy." If a record is found and retrieved by this key it is not possible to assume it has any other desired characteristics one may wish it to have. Therefore when a key is found with

"allergy" in the ring list of specialities, this provides a branching off point to search through a ring list of allergies. For example, if one wished to retrieve records that not only dealt with allergy but also immunology then the ring list of allergy would be searched till the key immunology was found. This hierarchy could be carried to any desired level and an appropriately desirable record retrieved. If more records were needed with the same characteristics the pointer of the retrieved item would be used to indicate the first record in the last nested ring that was searched. It is obvious that the elimination of searching all previous nested rings in the search would certainly facilitate the efficiency of retrieval in terms of time.

However, it is also obvious that the complexity of the ring structure imposes no small task on the programmer in originally setting up such a system. Furthermore, the problem of updating one's storage file would be almost as complex as the original program. Added to this is the costly disadvantage that such an organization requires added core space in order to handle all the pointers.

In concluding this section on file organization it should be noted that there are even more complex data structures based on the three main organizations of sequential, random, and list. Multilist, cellular multilist, indexed sequential, and tree organizations are but a few of the somewhat exotic techniques that have been developed. These techniques, however, are considered to be too costly and perhaps impossible to implement at the present time at this university. The reasons for this will be discussed in the first section of Chapter III. In all cases, however, one must weight, as Salton (1968, Ch. 7) has suggested, the

advantages and disadvantages of each file organization and also try to chose that data management design that is most suited to the demands of one's particular needs.

Data Coding Techniques

Data coding is a term applied to the transformation of data representations meaningful to the external world into a mode more suitable for machine processing. It is usually done to reduce space needed to store data or to provide a more suitable statistical distribution of terms in the store. In most information and retrieval systems, some sort of comparison is made between a user's request and these coded records. A certain degree of favorable comparison will retrieve that record for the user, hopefully meeting his needs. However, as it has already been pointed out (cf. p. 4) no retrieval system is able to be all things to all people. From past interests, however, reasonable objectives can be set.

Once these objectives have been determined, a match strategy can also be specified. This strategy, so chosen, will in turn determine how one will code his data. In general there are two widely used techniques for coding. The first is an *a priori* approach in which records entering store have been manually coded. The other is to turn the indexing over to the computer for automatic coding. These two techniques and the ramifications of their use will now be discussed.

Manual Indexing

Lipetz (1966) points out that "Satisfactory comparison . . . requires the ability to recognize the important features in the word. This is not an easy task to turn over to a machine [p. 177]." Abelson (1968, p. 419) agrees with this point of view, emphasizing the need for human judgment in information retrieval. He feels that professionals in individual fields of scientific research are essential custodians of knowledge who cannot be replaced by archives of any kind. It follows that those who feel this way would have more confidence in a system which retrieves records which were indexed manually before entering core.

The manual indexing procedure usually follows three basic steps. Firstly, the intended user and system designer decide upon criteria needed for retrieval. Secondly, a code is made of these criteria, and thirdly a user who is a specialist in his subject area, transcribes all records that will be put into store according to the code of the criteria. Later the user simply specifies which particular criteria he is looking for in a record and a search is made of the coded records to retrieve pertinent records. Altmann (1966, pp. 154-157) describes such design in the implementation of his system.

Indexing performed by trained indexers is extremely detailed and as such it is on the whole superior to automatic indexing. The difficulty of such indexing is the time and cost of indexing by a person who is not only trained in indexing but also a specialist in his own subject field. Though he would be required to code the original data and all updated versions, such a person would not be gainfully employed by indexing alone unless the bank was very large. Hence, such procedures require the

finding of a person who is interested in working in interdisciplinary fields. This of course is not always easy.

Automatic indexing

Since the advent of the modern computer, system designers have always dreamed of systems which would eliminate almost all human effort. Thus, attempts naturally were made to enable the computer to take over the task of indexing. The automatic indexing procedure follows only two basic steps. Firstly, the intended user and system designer decide upon criteria needed for retrieval; this step is the same for both manual and automatic indexing. Secondly, the programmer must write a program to assign indexes to records. This is usually accomplished by some kind of word matching; for example, author names, titles, phrases, citations, and so on are searched by the computer and if a match is found with words written into the program, the computer is instructed to convert the record to a code for more convenient use. Thus a user simply specifies which criteria he wants a record to meet, for example, "neurology, McGill, graduate." The computer would convert these specifications to the same code as used for the records in store and then search for a match.

There are, however, serious limitations to such procedures. First, there is the problem of synonyms. The programmer must foresee the possibility that records dealing with "cats" belong in the same cluster as records dealing with "felines." He must also be able to accomodate for users who may not use either of these words but may instead specify "pussy" or "kitten." It is likely that such a user would be interested in the records dealing with "cats" or "felines" but if the program has not converted all words to the same code he will not get such documents.

The inclusion of a synonym dictionary not only presents the problem of incomprehensiveness and programming effort, but also that of requiring more core space. As one uses up space for the program to index records and user requests, space is lost that could be used for blocking records, pointers, or similar features in one's file organization. Thus the use of automatic indexing necessitates large and expensive memories, excessive programming, and slower operation--if it can be done at all. Certain authors (Hammond, 1964, pp. 237-293; Wallace, 1964, pp. 225-235) also point to the difficulty of analyzing phrases and syntactic relationships in automatic indexing. They found that different subject areas not only had a unique vocabulary but that different habits were found among writers in using the most common words. Thus variation in subject field and variation in style of writing imposes serious restrictions on adequate automatic indexing. It seems, therefore, that Lipetz (1966) makes a valid claim in stating ". . . when concepts buried in multiple-word phrases must be recognized . . . the human specialist is still quite able to compete with the computer [pp. 155-156]." Garfield (1964, pp. 189-192) also claims that considerable standardization of stored documents is necessary before automatic indexing becomes adequate and this, he feels, is unachievable for many years to come.

Salton (1968, p. 345), however, has found in his work that automatic processing was not substantially inferior to manual coding. There is also the great convenience of eliminating the manual indexing step which is time consuming and costly.

The resolution of the question as to which is better will probably not be found until as Jones (1969, p. 32) has suggested, further experiments are carried out for the comparison of manual and automatic thesauri, thereby establishing unequivocally whether one is better than the other.

Associative Techniques

Tinker (1966, pp. 96-102) has shown that as more descriptors or indexes are assigned to a document and are required in a request, the more difficult it is to retrieve an item. Furthermore, it has been generally found that most systems stores rarely are able to give the user the items with the exact characteristic for which he is looking, even when limited descriptors are used. As a result, statistical association techniques have become widely used as a means of increasing the number of relevant records retrievable in response to a specific search request. It becomes possible, therefore, to retrieve not only those items which are an exact match of the criteria specified, but also, if needed, those stored items that meet most, but not all, of the criteria.

Doyle (1964, pp. 15-24) retrieves items on the basis of a ranking hierarchy, where items meeting the most criteria are obtained first, then successively those items that meet less and less of the criteria. Other designers (Wittman and Ingerman, 1967; Mathews & Thomson, 1967) use a threshold selection technique in their systems. Consecutive integers are assigned as weights, commencing with unity for the least perferred term, with a minimum value specified for a stored item in order for it to be retrieved. If the sum of the weights of criteria met

by a given record exceeds this threshold score it is considered pertinent. Salton (1968, p. 140) presents experimental evidence that the use of these weighted identifiers is always more effective than methods retrieving only records of exact matches.

Some researchers (Edmundson, 1964; Kuhns, 1964) use correlation coefficients as a means of ranking the retrieved items. A binary vector indicating the criteria is correlated with a similar vector for the stored item. Records are then ranked according to the size of the correlation coefficient and are retrieved in rank order.

Some ambitious attempts have been made to implement procedures that will retrieve items that do not belong exclusively to one category. Multi-dimensional approaches such as discriminant analysis by Williams (1964) and factor analysis by Borko (1964) have been developed to determine the degree of relevance of a given stored document with respect to more than one category.

Some of the most successful systems try to improve user's satisfaction with his retrieval system by providing iterative steps. Salton (1968, p. 345) has found that, in actual practice, users have many differing needs, some wanting very exhaustive answers and others being content with a single reference. One of the easiest ways to adapt to these various needs is a provision in the system for more interaction between the user and the machine. Many (Lipetz, 1966; Parker, 1966; Bryant, 1966; Licklider, 1965) suggest that results of initial searches should be given to the user. He then modifies his request and a more refined search is made on the basis of these needs. This iterative procedure, though somewhat slower, no doubt offers more satisfaction to the user.

As an extension of, and as a means of improving the time lag of the iterative approach, promising investigations (King, 1968; Rubenoff & Bergman, 1968; Belz, 1967) are being made with the so-called interactive systems. These systems, incorporating on-line terminals, are designed to improve search and retrieval by reducing the time between iterative searches. The user is provided with a terminal (for example, a cathode ray tube for display of retrieved items, and a typewriter for user responses) with which, in one sitting, he is able to see the results of his first request and make adjustments to his requests on the basis of information received, continuing the iteration until he is satisfied.

Some Significant Retrieval Systems

A review of the literature would be incomplete without a description of some of the more advanced and well known retrieval systems. One of the more widely accepted systems for retrieving documents has been the KWIC (Key Word in Context) system. Input is either running text, abstracts, author-title, or keywords. The system is generally used to index documents by the words in their titles. The computer reads all the words in all the titles and alphabetizes these words. Then it prints out these words for all titles in alphabetical order on successive lines but keeps with these words the content of the titles and a code for locating the full text. No attempt is made to associate synonyms so the user faces the task of searching for synonyms himself. To eliminate useless printing and searching, the system can be instructed not to print entries for common place words such as "and," "the," etc. The journal _Chemical Titles_ is such an index and serves as a alerting service for

chemists regarding recent articles that have appeared in selected journals. Many researchers (Benson, 1965; Stiles, 1965; Sage, 1965; Sprague 1965; Stewart, 1966; Stevens, 1964, p. 283) incorporate many of the basic principles of KWIC in their systems.

The limitations of such a system are, of course, not only the limitations associated with any automatic indexing system but also those imposed by the lack of a synonym dictionary.

Since this study is specifically related to medical applications, the MEDLARS (Medical Literature Retrieval and Analysis) system will be specifically noted. MEDLARS is a computerized information retrieval system for use by the National Library of Medicine, its design and implementation being carried out by General Electric, using similar techniques as KWIC. An entry into this system consists of a citation and its associated tags, described as a unit record. Both periodical and non-periodical articles may be a unit record. Kent (1966) maintains this is the most sophisticated and most conpletely automated medical library system available. However, Cookley (1967) points out that many of the titles of articles were often inadequate in reflecting the content of the article. Minker (1969) also criticizes the system in terms of its failure to reject irrelevant material. Therefore, despite its sophistication, MEDLARS still has the inherent weaknesses of many other automatic indexing systems.

One of the finest retrieval systems in the world today is the SMART system. Implemented on both the IBM 360 and 7094, it is essentially a laboratory for retrieval principles and procedures. Doyle (1969) refers to it as ". . . a <u>tour de force</u> in experimentation in documentation area the like of which is seldom seen [p. 271]."

Salton (1968, pp. 9-20) describes SMART as a fully automatic text processing system which manipulates documents and search requests, expressed in natural language, and produces the documents that appear to be most similar to requests. The system is characterized by several hundred different content analysis procedures, all available to generate indexes to records and requests, including word matching methods, stored dictionaries to lessen the effect of vocabulary variations, statistical and syntatic procedures to identify relations between words and concepts, and phrase generating methods. Thus a means is provided for attacking the content analysis process from a number of different viewpoints, each of which provides a somewhat different output. It is possible, therefore, for a search process to be conducted in such a way that search requests producing unsatisfactory results can be reproduced under altered conditions. The new output can be examined and, depending on requirements, further changes can be made until a satisfactory retrieval is obtained.

The user is also requested to identify those records considered to be most useful. The system then automatically adjusts the search request by increasing the weights of the requested terms that were also contained in the designated set of relevant documents and decreasing the weights of those not relevant. Effectively this process shifts the request vector so that it lies closer to the relevant document subset than to the nonrelevant subset. In this manner, similar future requests will presumably receive only the most pertinent records.

It would seem that this system has much to offer other designers in the structuring of their data management. If one were able to use the system for his own records he could compare which method provided the most pertinent records for the user and then incorporate that automatic indexing procedure into his own system.

Summary

This chapter has reviewed some of the literature of information retrieval as it is related to theory, system design, and data management. One can summarize the implications of these readings as follows.

There is certainly a need for the system designer to report his work in a very thorough, practical manner, noting all relevant specifications, problems, and evaluations of his work. Furthermore, his design should be analyzed in terms of the principles underlying the theory of information retrieval. This not only provides a basis for contributions in the field but also provides a frame of reference for evaluation.

The designer and programmer should also try to meet Soergel's (cf. p. 11) requirements for the programming language used in order to insure its maximum facility. The proper choice of a file organization is also necessary in order to meet the demands of the intended users as well as the available hardware. Close association with the intended users also facilitates the designer's understanding of objectives and expectation and thus enables him to choose the appropriate indexing procedure, implementation of a practical retrieval iteration, and adequate hierarchical outputs.

These implications as related specifically to this study will now be discussed in Chapter III.

CHAPTER III

STORAGE AND RETRIEVAL DESIGN

Introduction

This chapter confronts the problem of designing a specific system for handling multiple choice questions. In attempting to do this, Chapter three is divided into two basic units. The first unit deals with the generalized constraints and specifications that are expected to be encountered in designing retrieval systems for multiple choice items. The second unit deals with the implementation of this design, specifically that relating to medical documents.

Constraints and Specifications of System Design

Introduction

Before any information retrieval system is implemented there are a number of design constraints and specifications that must be considered. These are not exclusive to this study but apply to all information systems. Firstly, the designer must consider the nature of his data base, which invariably is determined by the discipline using the system. Secondly, the choice of file organization depends upon the available hardware and software. Thirdly, the designer must provide a system that is complatible with the human user. Lastly, the nature of a design will depend upon the available supporting resources, which includes not only manpower but also finances. The nature of these constraints will now be discussed in detail.

Data Base

Since the data base being considered is multiple choice examination questions, a designer must consider those constraints which are related to education and measurement. Multiple choice questions vary a great deal in their length; therefore the system design must provide for variable record lengths in retrievals. Furthermore, educators usually do not file only the text of an item, but also include information related to the item's subject matter, taxonomic level, correct response location(s), number of items used, year in which it was last used, the examination in which it was last used, the source of the item, its difficulty level, and biserial correlation. Some educators now provide audio-visual material with their questions, and this source is recorded. A design for storing and retrieving multiple choice questions on computer must therefore accommodate not only documents (the text of an item) but also measurements (an item's descriptive indexes). The design must incorporate the features of two standard retrieval systems--information retrieval and reference retrieval (cf. p. 6). Since educators most often choose items for an examination on the basis of their descriptive indexes the designer must also provide codes for these indexes either manually or automatically. Most of the descriptive indexes mentioned above are almost impossible to code automatically. Biserial coefficients and difficulty levels can only be known after an item has been used and analyzed, the procedures for which are entirely separate from retrieval. Similar problems exist for coding the number of times a question is used or in what examination it was last used. Taxonomic levels are most easily determined by the educator, not a programmer. Such conditions make it necessary for manual

coding to be used in lieu of automatic procedures for a design handling multiple choice items as its data base.

Hardware and Software

Available computer facilities and the programming language used to execute these features influence the design of such a system. At the University of Alberta the IBM 360/67 installation provides each user with 200,000 bytes of core storage, a relatively large space which would accommodate the large dictionary files used in random and list file organizations. However, the hardware facilities at this installation have not been completely debugged for handling these two latter files. Sequential files, on the other hand, do not present such problems and are therefore the safest to use.

Fortran IV programming language imposes no inherent constraints on the use of a sequential file and is therefore appropriate for such file organizations. This software does require, however, that alphanumeric characters be read under A (alphanumeric) format only. A checking procedure is then necessary in the design to insure that the text of an item is not read under the I (integer) or F (floating point) formats used to read an item's indexes.

The choice of a sequential file organization and the use of Fortran programming language in designing a retrieval system provides a means for the system to be widely adaptable. Firstly, the software is a widely known and used programming language in retrieval systems. Secondly, sequential file organizations are widely known, used, and easily debugged. Lastly, sequential searches require minimal amounts of core storage and therefore allow the system to be implemented on very small installations.

Human User

Any design must incorporate means to detect, diagnose, and if possible reduce the inevitable element of human error. This type of error is not localized to the intended users--educators. It is introduced by everyone connected with the system--writers, typists, coders, keypunchers, programmers, and users. Further constraints are imposed upon the system by educators, however, since optimal use of the retrieved items or data is only attained when the system accomodates the idiosyncracies and needs of users.

To reduce human error, the designer must provide checks to insure that records containing the text of an item and its indices are in proper order before being stored on tape. Provision must be made for correcting an item's text or its indexes, whether it be due to human error or the accumulation of new knowledge. Whenever and wherever the element of human mistake may be present, the designer must try to provide means to circumvent and/or give appropriate diagnostic messages to the user so that he may rectify the problems before making subsequent requests.

To increase the useability of retrievals, items should be ranked in order of their relevance to a user's request. However, professional educators rarely want a machine to choose only the exact number of items that will appear on an examination; therefore, a provision for giving an overflow of items of equal relevance is necessary in such a retrieval design, with checklists to facilitate decision by the user regarding which items are most suitable. Provision must also be made for educators who wish to modify their requests after primary retrievals. Since the usefulness of some multiple choice questions depends on proper syntax,

it is imperative that retrievals be in a form that could be directly copied onto an examination paper; the use of truncated words and omission of participles and conjunctions is not advisable.

In practice the designer must be well acquainted with the discipline of test design and with the intended users, educators, in order to anticipate the constraints imposed from these sources upon the design of a retrieval system for multiple choice items.

Supporting Resources

Before a design can be implemented one must have the available resources, both physical and financial. In addition to the manpower needed in a noncomputerized file of multiple choice items, a computerized system requires a keypuncher, a programmer, and a system designer. Writers, typists, coders, and test committees are of course common to both the manual and machine files.

An additional major source of expense, related to the computer itself, is incurred with a machine based system. Financial resources must be available not only for computer cards and tape, but also for the running time of and the amount of core used in the computer. Only the last two costs--computer time and space--are directly under the control of the designer. By using sequential files and incorporating proper job control language (such as blocking) one can keep costs to a minimum. Since education costs are rising the last constraint becomes of primary importance.

Summary

The left-hand side of Figure 1 flowcharts the design steps necessary to handle all the constraints considered. The very nature of this design for storing and retrieving multiple choice items requires that the designer be acquainted not only with the disciplines of information and computing science, but also with the disciplines of education and test development. Without an interdisciplinary approach it would be difficult, if not impossible, to design a system that would not only operate, but also meet the needs of those who would use it.

Discussion will now deal with the actual implementation of these specifications, the steps of which are outlined on the right-hand side of Figure 1.

Implementation of System Design

Introduction

In order to show the feasibility of the general design specifications in Figure 1, this study developed a retrieval system specifically for multiple choice items in medicine. This unit deals with that application. As each design specification is implemented, the reader should refer to Figure 1 in order to see the relationship between the general specification and this study's example of its specific implementation. The study incorporated a sequential file organization for storage and retrieval, the choice of which being determined by the following. Firstly, and as noted before (cf. p. 33), the IBM 360/67 at the University of Alberta is best suited to the handling of sequential files. Secondly, because the intended users wanted a thorough retrieval based on a variety

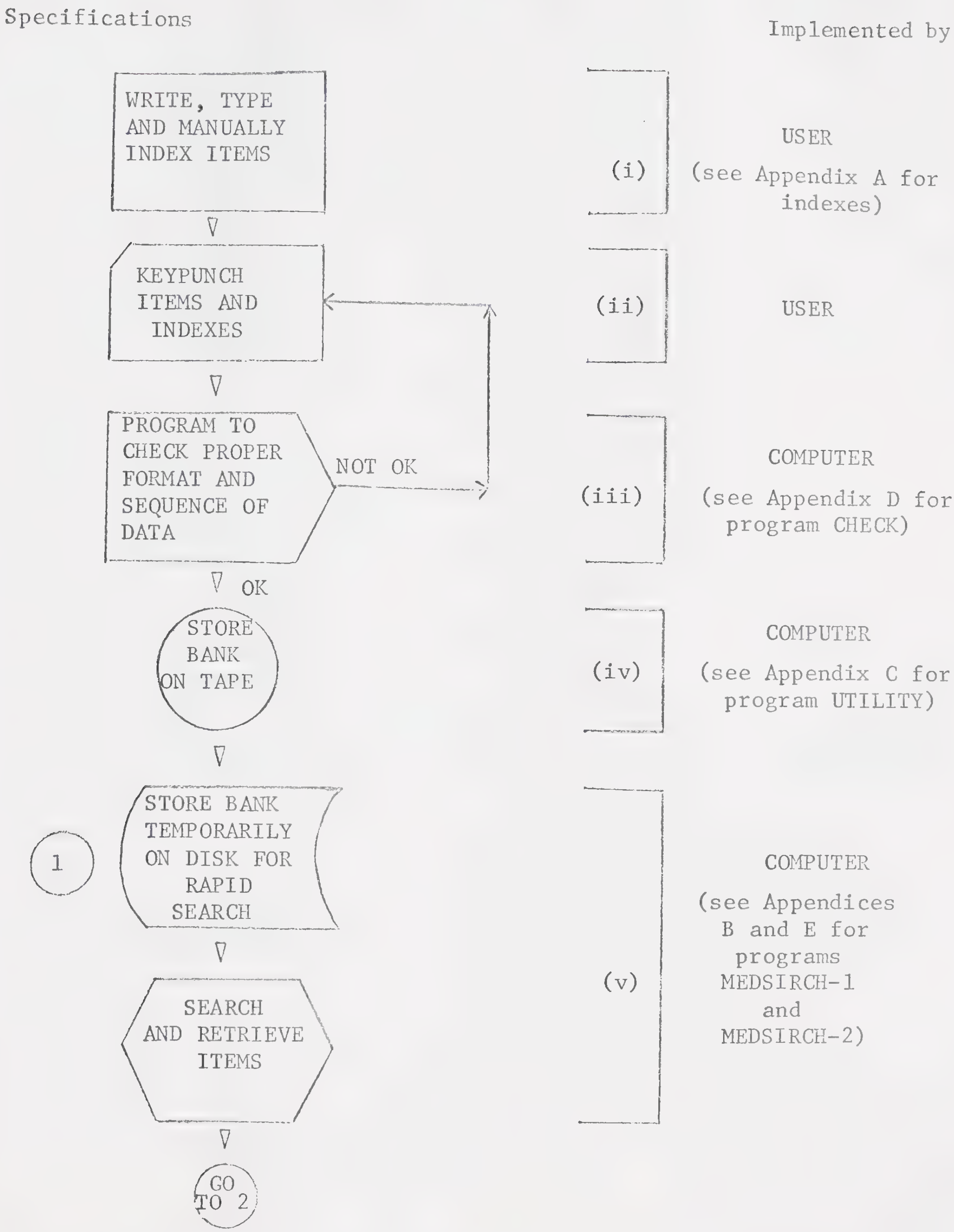

Figure 1: General Design Specifications for Use in Item Retrieval

Specifications

Implemented by

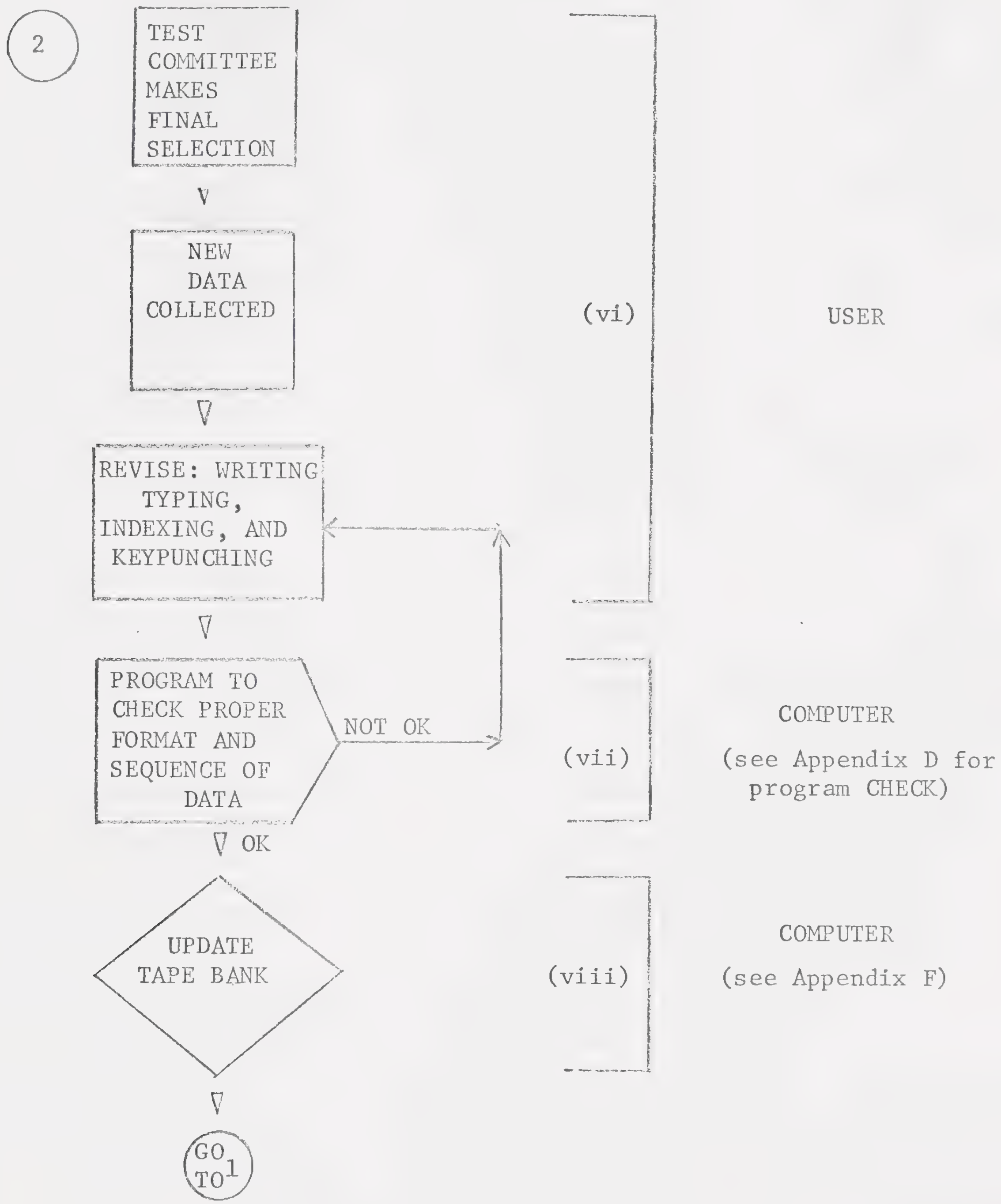

Figure 1: General Design Specifications for Use in Item Retrieval

of keys, Salton's suggestion (cf. p. 13) that a sequential file may be the most practical organization seemed valid. Thirdly, cost, in terms of the amount of core storage, is considerably lower. The system analysts at this university have indicated that in the immediate future users will be charged for increasing amounts of core in an exponential manner, for example, using twice as much core may be four times as costly. Since random and list files involve considerable amounts of active memory for dictionaries and pointers, it becomes immediately questionable whether or not the assumed increase in economy of time would out weight the added cost of core. Fourthly, the amount of decreased efficiency in use of time for a sequential organization for this study remains hypothetical. Remember that Dodd (cf. p. 16) noted that random access is not suited to rapidly accessing a large number of retrievals, and that retrieved items are usually of a uniform length. Though Dodd has not been specific as to how many retrievals is classified as a "large number," and thereby estimate such an effect in this study, this investigator did know that stored items varied a great deal in the number of records composing each document. Conversely, blocking is available in the sequential file which, as shall be noted later in this chapter (cf. pp. 74-76), increases the efficiency of processing such a file by factor approximating the blocking factor. Therefore, only further investigation comparing the use of sequential, random, and list files for data used would be able to tell which organization would be optimally efficient as well as inexpensive.

Having chosen a sequential type of file organization a decision was made to which coding technique--manual or automatic--was to be used. To provide a frame of reference within which a comparison could be made of

these two techniques, the Department of Medicine was consulted as to the characteristics they would be looking for in a retrieved item. They expressed these four elements as most desirable: (a) area of subspeciality (such as, allergy, cardiovascular, collagen diseases, etc., (b) type of question (single or multiple answer), (c) taxonomic level (approximating Bloom's (1956) taxonomy), and (d) core level (that is, whether a question covered essential, important, or not so important medical knowledge.

Three of these specifications--subspeciality, taxonomy, and core--are definitely "concepts found in multiple-word phrases" as Liptez (cf. p. 24) would say, and as such are most difficult, if not impossible, to code automatically by the use of computer. Any attempt at such automatic coding would require an enormous dictionary that not only included synonyms for words and phrases but also procedures for analyzing phrases and syntatic relationships. The programmer would be required to know the content of each document and the linguistic devices used to carry its meaning. It was this investigator's conclusion that the task would be difficult, if not impossible, to make any attempt at automatically indexing multiple choice questions under these codes seem futile. For support of this viewpoint one is referred to Kurtzke (1967) who states "I do not believe that standard medical records will be amenable to computer search and retrieval except with the imposition of a trained human encoder [p. 128]."

Little contribution was gained in the area of indexing, therefore, from such reputable systems such as KWIC, MEDLARS, and SMART. Note, however, that some of the major criticisms levelled against these systems were directly related to the shortcomings of inadequate, automatic

indexing. By choosing to use a manual indexing technique, this design became less convenient but, as shall be seen in Chapter IV, can provide more pertinent retrievals.

Item Indexes

Since each multiple choice item was to be manually coded it was necessary to provide the human encoder with a standardized format for the indexes which would accompany each item in the bank. The reader is referred to Appendix A for the listing and format of these indexes (which is a specific implementation of section (i) in Figure 1). The codes used were determined by consultation with the Department of Internal Medicine, the format by programming specifications. All item indexes can be specified on two cards, with each code being alloted to specific columns on one of these two cards. One of the rigidities encountered in the use of computers is the requirement that codes be allotted to the exact columns specified or else the information will be interpreted incorrectly by the computer.

The following is an explanation of the indexes, that is the type of information desired by this particular medical department for each of their examination questions. As indicated before (cf. p. 25) only a limited number of criteria should be made in a request in order to improve retrieval; four criteria were used for this study.

The first of these four was "area of subspeciality." The term _subspeciality_ implies a category of a larger speciality, and indeed that is what this category is. Within the speciality of Internal Medicine, particular areas of medicine are delimited for purposes of education, not unlike sections taught in a Science or Social Studies course. To be

able to evaluate a candidate comprehensively an adequate representation, if not all, of these defined subspecialities were needed. Twenty-three areas were specified--allergy, cardiovascular, collagen disease, etc., through to physiology--and represented by the codes 1, 2, 3, ..., 23 respectively. This index was to be punched in an item's first parameter card in columns one and two.

The seond criterion, termed "type of question," referred to the two types of questions used by this medical department--single and multiple answer--both of which are explained by Hubbard and Clemans (1961). The multiple answer question (as distinct from the single answer question having only one of five choices correct) has two or more alternatives correct. If an item being coded was a single answer type, a one was to be punched in column three of the first parameter card, if multiple answer, a two.

The third criterion was to be punched in column four and referred to the "taxonomic level" of the item being coded. The taxonomy is a cruder classification than Bloom's (1956) hierarchy for the cognitive domain, a one referring to a classification termed factual, a two for comprehension, and a three for problem solving. The first two were analogous to the first two levels of Bloom's hierarchy--knowledge and comprehension; the last, problem solving, encompassed the remaining hierarchy as suggested by Bloom with the exception that little or no attempt was made in composing multiple choice questions at the level of synthesis.

The fourth and final criterion was for classifying the importance of the subject matter tested in a given item. Called "core level" a one, two, and three refer to essential, more important than unimportant, and more unimportant than important material respectively. It was felt that this category was desirable in evaluating the comprehensiveness of a particular group's or candidates's knowledge.

There are other indexes accompanying each item and the reader may refer to Appendix A to note their characteristics, codes, columns and parameter card number in which the codes were punched, and the abbreviations which the computer used in designating a particular item's codes.

In summary these are the remaining indexes which, although not used in retrieval, provided the user with further useful information, and presumably aided him in modifying his requests for iterative searches and his own final selection. (a) "Second area of subspeciality" allowed an item's inclusion in more than one subspeciality category if necessary. (b) "Source" indicated the institute or country from which the item was obtained. (c) "Province": if the item was authored in Canada (number three in "source"), this index was to indicate the medical institution from which the item was obtained. (d) "Audio-Visual" code was an optional code for indicating that the item had additional material to that of the written text of a question. Since no hardware facilities exist for storing line graphs, photographs, color sequences, slides, movies, or video material on the 360/67, items making use of such material had to be identified. (e) "Audio-Visual I.D. location" would have been specified also if the previous code, "Audio-Visual," had been specified. This identification number would indicate to the user where this extra material was located in the audio-visual file. At present Internal

Medicine is not using questions incorporating audio-visual material. This provision was incorporated in anticipation of their future use. (f) In the appropriate column(s) 15-19 a one was to be punched indicating the "correct response alternative(s). For example, a single answer question with choice three correct would have a one punched in column 17; for a multiple answer question with choices one, two and five correct, a one would be punched in columns 15, 16, and 19 respectively. (g) "Language" indicates whether or not the item is available in English, French, or both languages. Since the examinations set for entrance to the Royal College of Physicians and Surgeons can be taken in both languages, the code is a meaningful entry. (h) "Number of times used" specified the number of times this particular item had been used for the Royal College's examination. If this has been left blank no further entries were required in the remaining columns of parameter cards one or two except for the speciality, the card and the item identification numbers. These numbers must be specified on both parameter cards, as will be explained later. (i) "Last year question used" required only the last two digits of the year. (j) The next four codes referred to the last examination in which a particular item was used--"Number of question on last exam," "Graduate or undergraduate exam," "National or local exam," "ID of exam," "Number of examinees on last exam." Though there was no storage of questions for local or undergraduate examinations a provision was made to accomodate any future moves to use an information and retrieval system at these levels. (k) In columns 35-36 and 37-38 the "'p' values" or difficulty levels of a single-answer item were to be specified. Provision was made for only the last two testings of this item, as the examiners felt that no item would be used, or at least left

unmodified, after three testings. (l) Columns 39-40 and 41-42 allowed specification of the biserial coefficients for the same testings of an item. (m) Columns 35-42 were of course not applicable for multiple-answer items and would be left blank for multiple-answer questions. Provisions for item analysis data concerning multiple-answer type of questions involve more detail. Since each choice received a mark (a candidate marked it as correct or incorrect and received a mark for the correct identification), each had a difficulty level and biserial coefficient. Added to this was total difficulty of the item (average of the difficulty levels of each of the five choices) and the biserial coefficient of correlation for the total item. Columns 43-66 of parameter card one allowed for this data for the last recorded testing year, and columns 1-24 of parameter card two for the second last recorded testing year. (n) In columns 25-34 of parameter card two, specifications were made for "Proportion of last test selecting . . ." choice one to five. (o) Finally in columns 74-75 provision was made for the use this coding technique by other specialties. For Internal Medicine the code "01" was used. Any other specialities joining would receive another distinctive number. Column 76 indicated the parameter card number, one and two for the first and second parameter cards respectively. Columns 77-80 were provided for the identification number of the item, each item in the blank receiving a unique value up to a maximum of 9999.

In concluding this section on item indexes, the reader will note that the codes referred to the specific needs expressed by the Department of Internal Medicines. The numbers (or codes) themselves, however, are not inherently restricted to a particular subject matter field. They

are simply a means for reducing information as known to the human into a more concise, usable form for the information machine.

Programming for Retrieval: MEDSIRCH-1

To provide an iterative approach to retrieval two programs were written in Fortran IV programming language and termed MEDSIRCH-1 and MEDSIRCH-2. The first of these programs provided the initial search, the second being used for the modified requests. In the following explanation of these programs the reader will see how programming effort can be reduced by attempting to use many of the same features in both programs.

Appendix B provides the compilation listing of this program. Provided stored items and indexes and user requests conform to format specifications, this listing of MEDSIRCH-1 provides the Department of Internal Medicine with a debugged, effective, functional program for retrieving multiple choice items.

Preparation of items for storage. To implement section (ii) of Figure 1, this study required that all items be punched according to rigid format specifications. Specifically the text of an item had to be restricted to columns 2-69 of each punched card. Since the total item often consists of many lines, provision was made for indicating the item identification number on all cards. This number was punched in columns 74-77. The last column (80) of each card containing a given item needed one of three indexes--"C", "*," or a blank. These codes have the following functions. A "*" indicates that when an item is retrieved the printer should skip a line before printing the text stored on the next

card. A "C" causes the printer to write on the next line. This spacing provides an aesthetic quality to the output and presents the text in a manner similar to that used in an examination. If column 80 has a blank, this indicates that it is the last card containing the text of a given item. The program then instructs the computer to switch from the format for reading alphanumeric text (A format) to a format for reading the two parameter cards containing an item's indexes (Appendix A).

Having read an item and its two parameter cards the program then expects the appearance of another item with its parameter cards. Therefore, in setting up the bank one must punch an item, its parameter cards, the next item, its parameter cards, and so on to the end of the bank.

The end of a bank is designated by an "E" in column 80. MEDSIRCH-1 then instructs the computer to stop reading the bank.

The bank of items for this study was stacked on tape (section (iv) of Figure 1) with the utility program provided by the IBM 360/67 system. The system cards for calling this facility are given in Appendix C. This type of data set is read in the same manner as cards, but can contain a greater number of records than the limit of 2000 records imposed on cards. Though 2000 cards is an arbitrary number decided upon by the operators of the IBM 360/67 at this university, one would find it most difficult to manually use any more records than this in any case.

Before cards were stacked onto tape this investigator found it advantageous to use the program CHECK (Appendix D) to insure that items and their respective parameter cards were properly collated (section (iii) of Figure 1). This was necessary for a number of reasons. (_a_) Firstly, and most importantly, if any card (other than the last card) containing the text of the item had a blank in column 80 the computer ceased

executing MEDSIRCH-1 after reading such a card. This was due to the fact that a blank index in column 80 caused the computer to expect a parameter card to be read under integer and floating point format. Since the next card was still in alphanumeric format, a syntax error is registered and execution is arrested after printout of the message IHC215I--illegal decimal character. (b) Secondly, the program CHECK indicated a number of errors that the author or the key puncher may have made in collating the items and their parameter cards. Table 1 provides a list of possible mistakes the user may make and the diagnostic messages the program CHECK gives for each one.

The explanation of the CHECK program is included in this subsection since it is also used as the PDISC subroutine of the program MEDSIRCH-1. PDISC is called from the mainline program in order to write the entire stacked tape onto a more efficient medium, that of disk (section (v) of Figure 1). This procedure is done for two reasons. (a) Disk as opposed to tape, is a direct access medium and as such requires no rewind time. Instructions to REWIND this type of data set involve the simple movement of a READ/WRITE head (a hardware device) from its present location to the beginning of a file. Conversely, a REWIND instruction to tape literally involves the rewinding of a magnetic tape, not unlike the procedure used on tape recorders. The use of disk, therefore, involves great savings in time, especially if REWIND instructions are used as much as is done in MEDSIRCH-1. (b) Writing on disk is done without format and as a result the information is in a form that is immediately processable by the computer.

TABLE 1

DIAGNOSISING HUMAN ERROR

Human Error in Creating Bank	Example of Diagnostic Messages
All 1D numbers for an item are not the same.	****NOTE,NOTE,NOTE,NOTE**** MISMATCH OF IDS WITHIN ITEM: 360 AND 3600
Card containing test of an item has unrecognizable character in column 80.	*****NOTE NOTE NOTE NOTE NOTE ITEM 83 HAS THIS INCORRECT ALPHABETIC ENDING: (D) ON CARD 9
Item ID number is not the same as accompanying parameter cards ID number.	****NOTE NOTE NOTE NOTE**** MISMATCH OF IDS BETWEEN ITEM AND PARAMETER CARD: ITEM ID IS 1044 PARAMETER ID is 1045
Second parameter card preceeds first parameter card	****NOTE NOTE NOTE NOTE PARAMETER CARDS ARE OUT OF ORDER FIRST PARAMETER CARD READS: CARD 2 ITEM # 74 SECOND PARAMETER CARD READS: CARD 1 ITEM # 74
Pair of parameter cards do not have same ID number.	****NOTE NOTE NOTE NOTE PARAMETER CARDS ARE OUT OF ORDER FIRST PARAMETER CARD READS CARD 1 ITEM # 12 SECOND PARAMETER CARD READS CARD 2 ITEM # 120

<u>Preparation of search requests</u>. The use of MEDSIRCH-1 requires minimal effort in the preparation of requests. The user is required to specify in fields of five the subspeciality, type of question, taxonomic level, and core level codes as criteria to be met by items retrieved. For the fifth code the user must also specify the minimal number of items required; the sixth and last specification--hierarchical level--indicates the level of pertinency to which retrieval may procede if the minimal number requested is not met. The six entries constitute a request. Though a user may make as many requests as he desires, each one is punched on a separate card. To indicate the end of his requests the user inserts a blank card behind the last request card.

Figure two shows a set of requests with the concluding card blank. The first card requests a retrieval to the last hierarchical level for a minimum of six items that meet these restrictions: (<u>a</u>) subspeciality: one (allergy), (<u>b</u>) type of question: two (multiple answer), (<u>c</u>) taxonomic level: one (factual), and (<u>d</u>) core level: one (essential information). After the program searches for such items in the bank, it then will search for items meeting the specifications on the second card and third card respectively. The blank card will properly stop the execution of this particular set of requests.

<u>Use of DATA statements</u>. The mainline of MEDSIRCH-1 and its subroutine PARMTR make valuable use of DATA statements in converting the numerical codes of the parameter cards (Appendix A) and request cards into verbal text that is written in the output. This verbal text is designated in Appendix A in parentheses to the left of each index. For example, a "1" for subspecialty will be printed in the output as ALL, referring to allergy.

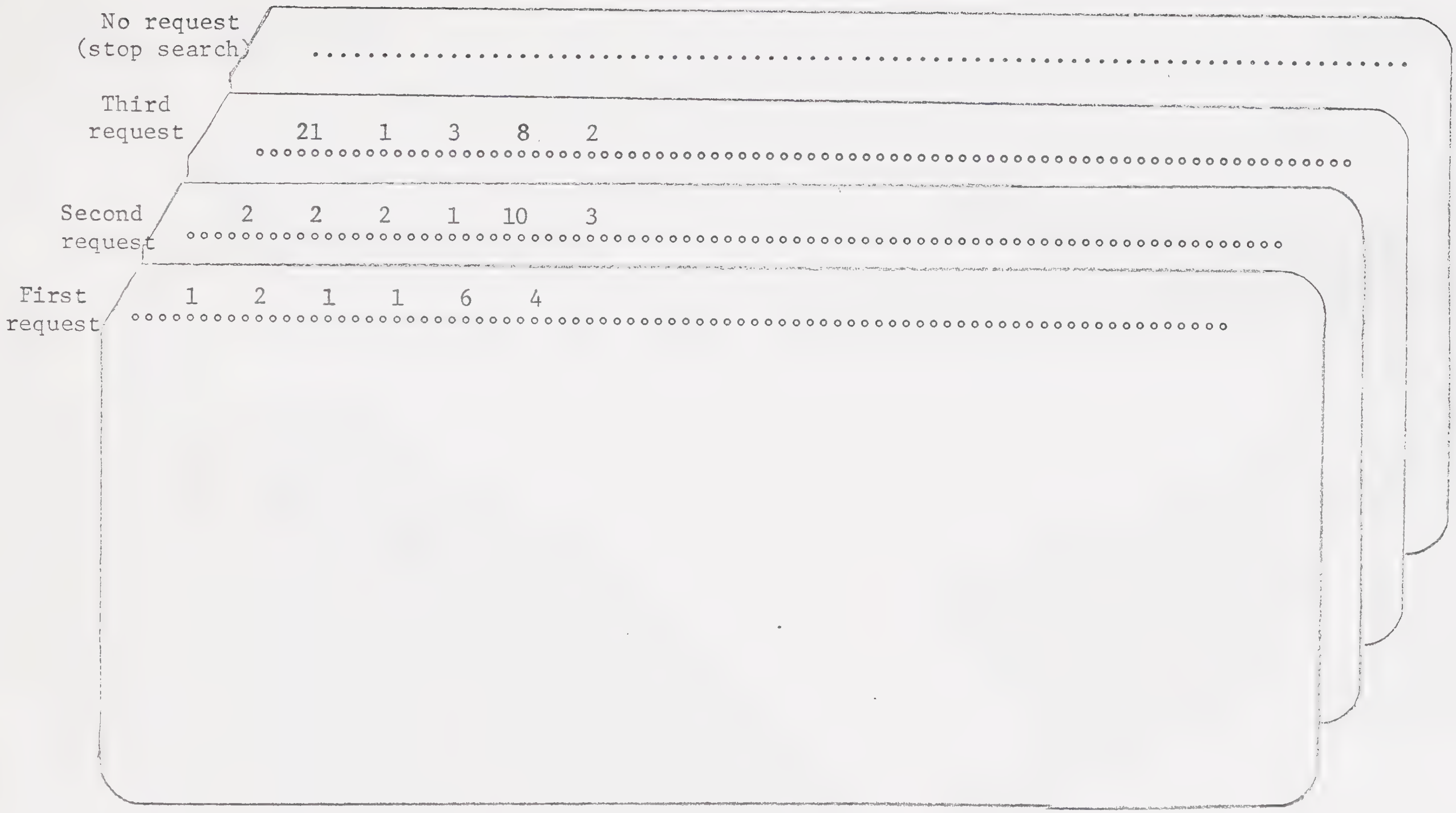

Figure 2. Set of Search Requests for MEDSIRCH-1

Use of DATA statments is also made for converting the codes placed in the 80th column of each card of the text of an item ("C," "*," "E," blank) into useable variable syntax as these alphanumeric characters are meaningless in themselves.

Search technique. Execution in MEDSIRCH-1 begins with the mainline calling the subrouting PDISC where the tape bank (data set eight) is subjected to the analyses of the program CHECK. The bank is then put on the direct access medium of disk without format and is thereafter referred to as data set one (see section (v) of Figure 1). After the entire bank has been transferred to this medium in PDISC, a return is made to the mainline where data set one is rewound in order to begin searching for all items pertinent to a user's request. At this point in execution MEDSIRCH-1 is at section (i) of Figure 3. This figure flowcharts in detail the search strategy used in this program. Search of the entire bank on disk is made to see how many, if any, of the four criteria specified in the request are met by each item. If a particular item does not meet any of the criteria it is ignored. If an item meets any criteria it and its parameter cards may be rewritten as other data sets, the location of which is determined by the following strategy for hierarchical output of retrieval items.

All items meeting the four criteria are written on data set six, that is, are printed in the output with a preceeding title indicating that the items meet all four criteria specified by the user; see section (iii) of Figure 3.

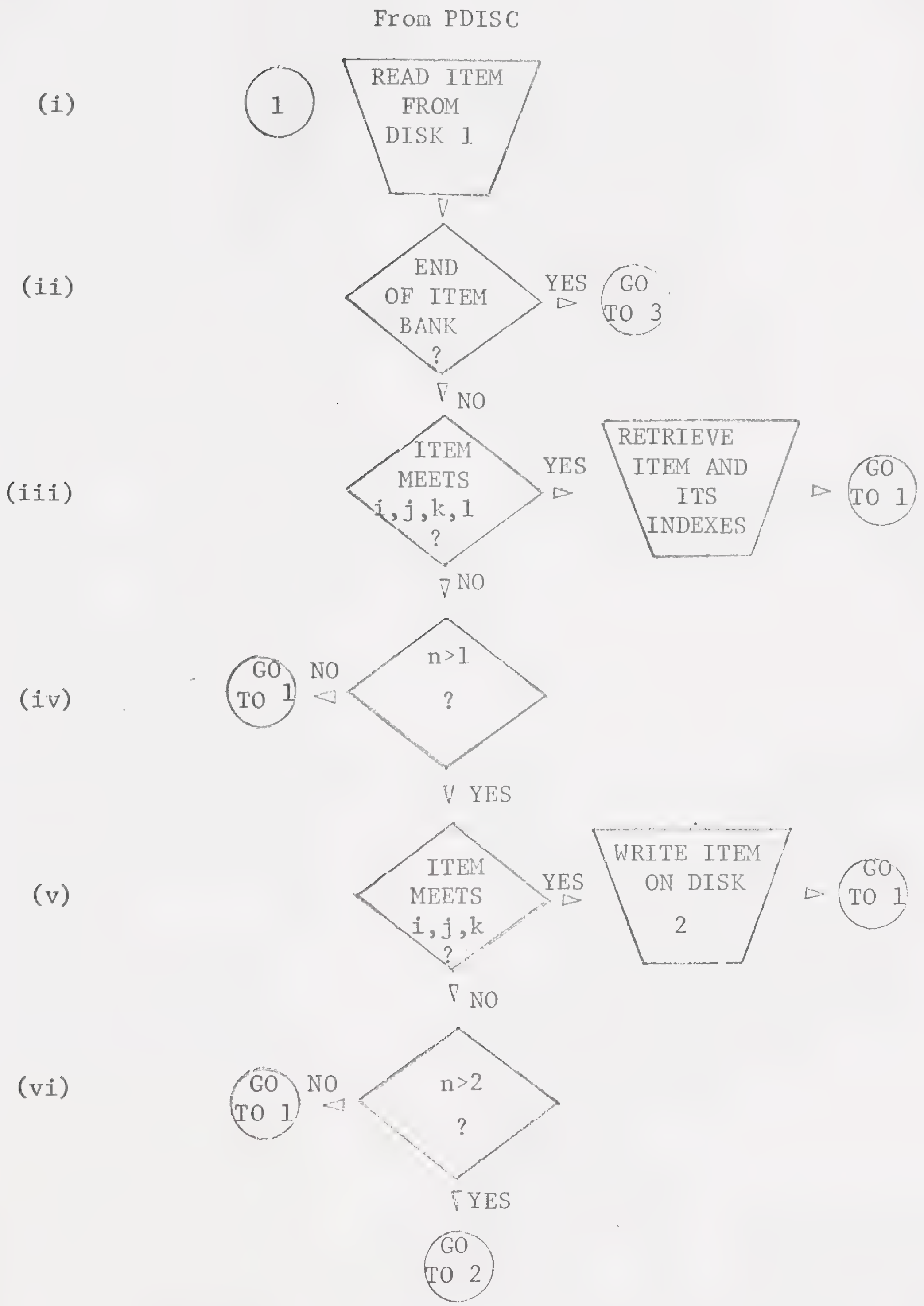

Figure 3: Search Strategy for User's Request i,j,k,l,m,n (MEDSIRCH-1)

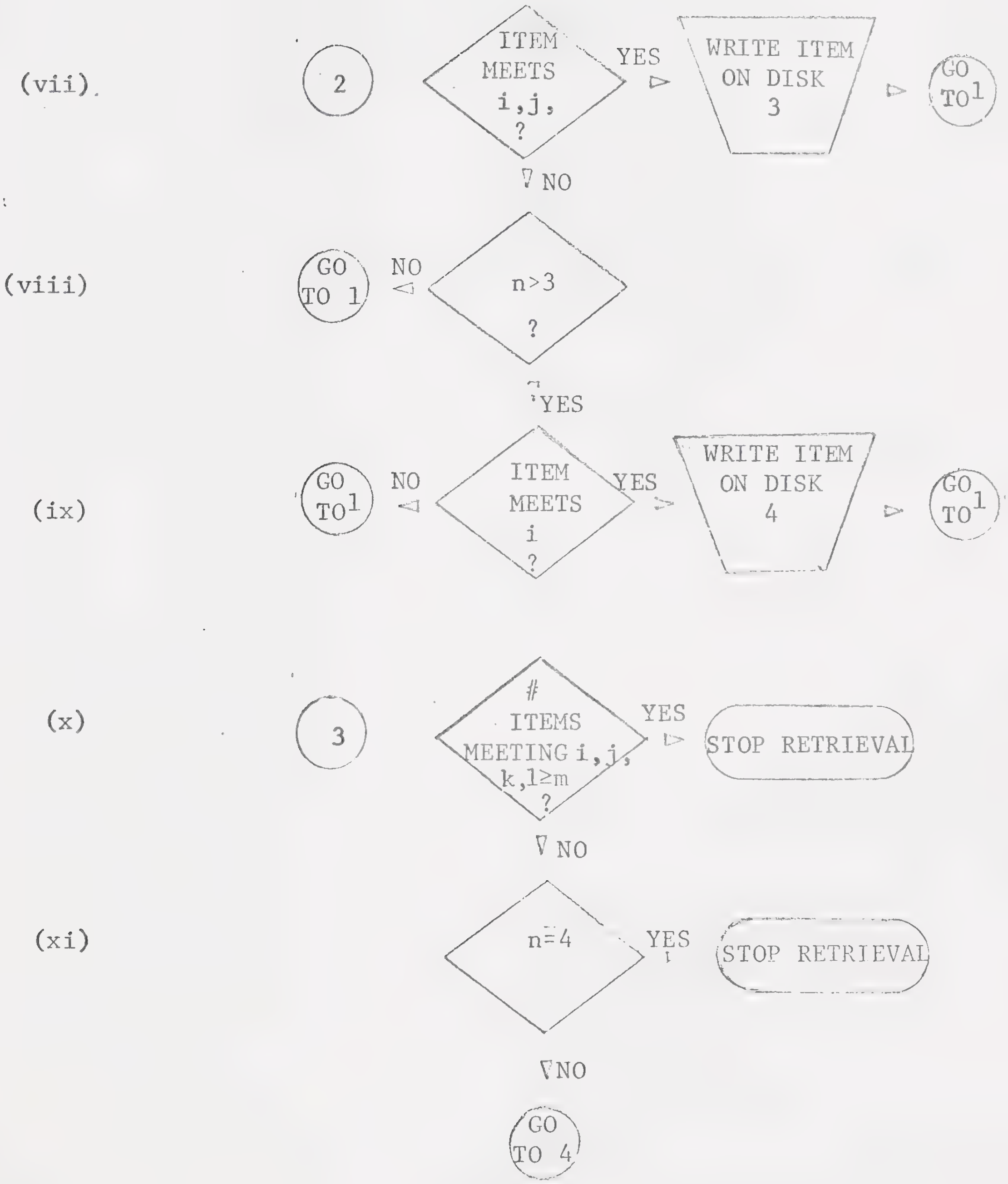

Figure 3: Search Strategy for User's Request i,j,k,l,m,n
(MEDSIRCH-1)

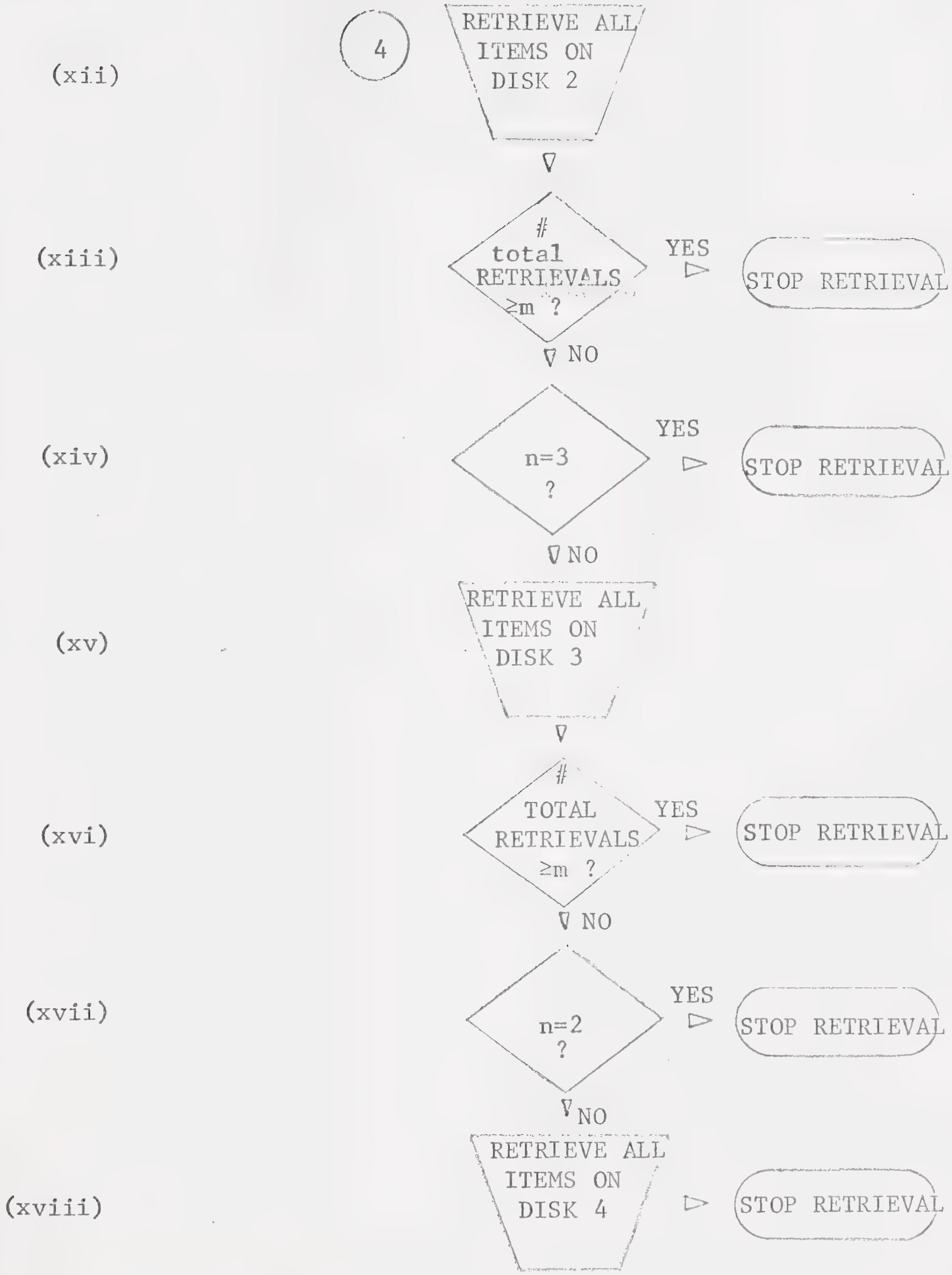

Figure 3: Search Strategy for User's Request i,j,k,l,m,n
(MEDSIRCH-1)

Those items meeting only the first three restrictions may be written onto another disk and referred to as data set two (section (v)). A similar procedure is carried out for those documents meeting the first two and one restriction where each set may be written on other disks and referred to as data set three and four respectively (sections (vii) and (ix)). Counters are kept for the number of items in each data set.

After the original bank (data set one) has been completely searched (section (ii)), a check is made as to the number of items that were printed (section (x)). If this number of pertinent retrievals is equal to or greater than the minimal number that the user requested, this particular search is terminated and a new set of requests is read. If this number of pertinent retrievals is less than the minimal number that the user requested then those items in data set two (those meeting only the first three restrictions) may be read and printed in the subroutine SORT (section (xii)). If the minimal number is still not met, or is exceeded, those items meeting the first two restrictions and, if necessary, one restriction, may be retrieved from the appropriate data sets (sections (xiii-xviii)). It is possible that a bank with too few items in a subspecialty may not provide the user with enough documents even at the last hierarchical level.

Such a strategy implicitly assumes that the four criteria are ranked relative to their importance. Thus, a user who is not able to receive enough items meeting all four criteria considers that the next most pertinent retrievals are those meeting <u>only the first three</u> criteria, <u>not any three</u> criteria. The analogous situation is true at lower hierarchical levels of pertinency. The choice of this strategy was based on the needs expressed by the intended users of this program,

the Department of Internal Medicine. Since the degree of success for any information system is improved by the extent to which the system accomodates the idiosyncracies of users, the choice of this search strategy for Medicine is valid. Similar needs exist in other educational fields as well. At examination time school teachers want items first and formost for a particular grade, then for a course (e.g., Social Studies), then for a particular section (e.g., geography), and at least within a particular range of difficulty levels and/or biserial coefficients. If items were retrieved in subsequent hierarchial levels on the basis of meeting any three, two, or one restriction, some retrieved items would be irrelevant. For example, consider a request for items in grade eight Social Studies, on geography, with difficulty levels within the range .2-.8. If items were retrieved at hierarchical levels on the basis of meeting a particular number of criteria , regardless of their relative rank order, retrievals for this request might be items with correct difficulty levels, in geography, but for the wrong grade. Presumably, therefore, the retrieval strategy is valid for most needs where order of restrictions is ranked.

It was discovered after the first retrieval that without the last specification--hierarchical levels--retrievals were often redundant. For example, suppose a request was made for six items with indexes 20-1-1-1 for which the bank could provide only three such items. The program would then automatically retrieve items with 20-1-1-2 and 20-1-1-3 in order to build up the retrieval number to six. The user may have specified in another request, however, a set of criteria exactly like those items retrieved at the second hierarchical level, for instance,

20-1-1-3. All items with these criteria as characteristics therefore would be retrieved twice.

In order to reduce this redundancy, the user's request was modified to include one more variable, the hierarchical level to which he wished a retrieval to be made. In this way he could determine the number of levels of hierarchy to which retrieval can take place, regardless of whether or not the number of items he hoped to receive was attained. By specifying "4" (section (iv) of Figure 3) he instructs the computer to retrieve only items meeting four restrictions (section (xi)); a "3" (section (vi)) indicates that retrieval cannot extend into any lower hierarchy than that in which items meeting only the first three restrictions are found (section (xiv)). Similar procedures apply for the specification of "2" (sections (viii) and (xvii)); a "1" would allow retrieval to take place to all four levels if necessary (section (ix) and (xviii)).

In summary this hierarchical specification may override the specification for "number of items desired." If, however, the user allows retrieval to proceed to a lower hierarchy than is needed, then the specification "number of items desired" terminates retrieval.

Increasing useability of output. An effort was made to increase the ease of reading retrievals and, hopefully, aiding the user in finding the most relevant retrievals.

Since retrievals may extend to varying hierarchical levels subroutine SORT instructs the printer to write appropriate messages to the user indicating how many restrictions are being met and whether or not a given level of the hierarchy is able to fulfill the user's request.

Table 2 gives a list of all the possible messages. Items having the characteristics designated by these messages follow immediately after in the printout. In order to keep track of the number of documents being printed out for a particular request, a cumulative number is printed beside the printed text of the item as well, and carries through all hierarchical levels. As a means of further identification the bank identification number of each item is also printed.

Since not all items have the same degree of relevance (even within a given hierarchical level) provision has been made so that each item and its parameter information is printed on a different page. If the user so wishes, he may easily tear out the most pertinent items since each computer page is perforated.

A listing for each request is printed as an indication to the reader as to which of his many requests a particular retrieval applies. A sample heading is provided in Table 3 for the request 21-2-1-2-1-4.

Though a search is performed on the basis of a document meeting only four of the indexes, any retrieval does include a readable printout of the remaining indexes where the human coder has been able to provide information. The subroutine PARMTR performs the conversion of indexes to a mode that is interpretable by the user. Included in Table 3 is an example of such a listing entitled PARAMETERS FOR THIS ITEM.

The subroutine PARMTR prints a check list for a person called the REVIEWER. The Department of Medicine makes use of such a person before a group, called the "test committee," finally selects which retrieved items will actually be used in an examination. This reviewer, in essence, performs an "updating" function, whereby any faults (grammar, content, etc.) found in an item may be noted before the final copy of the item

TABLE 2

HEADINGS INDICATING PERTINENCY OF RETRIEVALS

Condition	User Information Printed by Program
1. Items following have met all four restrictions submitted by the user.	THE FOLLOWING ITEMS MEET THE ABOVE RESTRICTIONS
2. Items are needed meeting only the first three restrictions in order to meet the number of items required.	THE ABOVE [2] ITEMS MET THE FOUR RESTRICTIONS SUBMITTED. THE NUMBER OF THESE ITEMS IS LESS THAN THE QUOTA ASKED. TO MEET QUOTA, ITEMS MEETING THE FIRST THREE RESTRICTIONS ARE GIVEN. WITH THIS SECOND SET OF ITEMS THE QUOTA IS MET OR EXCEEDED.
3. As in two except number of items required is still not met.	THE ABOVE [2] ITEMS MET THE FOUR RESTRICTIONS SUBMITTED. THE NUMBER OF THESE ITEMS IS LESS THAN THE QUOTA ASKED. TO MEET QUOTA, ITEMS MEETING THE FIRST THREE RESTRICTIONS ARE GIVEN. HOWEVER, THIS IS STILL LESS THAN THE QUOTA.
4. Items are needed meeting only the first two restrictions in order to meet the number of items required.	TO MEET QUOTA, ITEMS MEETING THE FIRST TWO ARE NOW ALSO GIVEN. WITH THIS THIRD SET OF ITEMS THE QUOTA IS MET OR EXCEEDED.

TABLE 2 (cont'd)

5. As in four except number of items required is still not met.	TO MEET QUOTA, ITEMS MEETING THE FIRST TWO ARE NOW ALSO GIVEN. HOWEVER, THIS IS AGAIN LESS THAN THE QUOTA.
6. Items are needed meeting only the first two restrictions in order to meet the number of items required.	TO MEET QUOTA, ITEMS MEETING THE FIRST RESTRICTION ARE NOW ALSO GIVEN. WITH THIS FOURTH SET OF ITEMS THE QUOTA IS MET OR EXCEEDED.
7. As in six except number of items required is still not met.	TO MEET QUOTA, ITEMS MEETING THE FIRST RESTRICTION ARE NOW ALSO GIVEN. HOWEVER, THE QUOTA CANNOT BE MET WITH THE PRESENT STORAGE OF ITEMS.

TABLE 3

SAMPLE RETRIEVAL TO REQUEST SPECIFIED AS 21-2-1-2-1-4

STRICTIONS IMPOSED:

AREA OF SUBSPECIALTY	21	- PATH
TYPE OF QUESTION	2	- MULT ANS
TAXONOMIC LEVEL	1	- FACT
CORE LEVEL	2	- IMP.
NUMBER OF ITEMS REQUESTED	1	
HIERARCHICAL LEVEL	4	

THE FOLLOWING ITEMS MEET THE ABOVE RESTRICTIONS:

1. A PATIENT HAS HAD A BIOPSY OF A CERVICAL LYMPH NODE. THE DIAGNOSIS OF HODGKIN'S DISEASE IS MADE. ITEM ID IS:1107

WHICH, IF ANY, OF THE FOLLOWING FACTORS ARE OF VALUE IN THIS CASE?

1. AGE OF THE PATIENT
2. SEX OF THE PATIENT
3. HISTOLOGICAL GRADE OF THE DISEASE
4. STAGE OF THE DISEASE
5. TIME ELAPSED FROM FIRST SYMPTOMS TO THE MAKING OF THE DIAGNOSIS

TABLE 3 (cont'd)

PARAMETERS FOR THIS ITEM:

AREA OF SUBSPECIALTY:	PATH
SECOND AREA OF SUBSPECIALTY:	HEMAT
TYPE OF QUESTION:	MULT ANS
TAXONOMIC LEVEL:	FACT
CORE LEVEL:	IMP.
SOURCE:	CAN
UNIVERSITY:	UWO

CORRECT RESPONSE ALTERNATIVE:

CHOICE 1	CHOICE 2	CHOICE 3	CHOICE 4	CHOICE 5
1	1	1	1	0

THE QUESTION IS AVAILABLE IN ENG.ONLY

NUMBER OF TIMES USED: 1
LAST YEAR QUESTION USED: 66
QUESTION NUMBER ON LAST EXAM: 9

TYPE OF EXAMINATION IN WHICH ITEM WAS USED: GRAD.
NAT. EXAM

DIFFICULTY LEVELS AND BISERIAL COEFFICIENTS OF THIS MULTIPLE-ANSWER TYPE OF QUESTION FOR THE LAST RECORDED TESTING YEAR:

	1ST CHOICE	2ND CHOICE	3RD CHOICE	4TH CHOICE	5TH CHOICE	TOTAL ITEM
'P'	0.35	0.30	0.60	0.75	0.50	0.50
'R'	0.20	0.35	0.26	0.29	0.41	0.37

REVIEWER PLEASE COMMENT ON:

CONTENT	OK?....
SENSE	OK?....
GRAMMAR	OK?....
RESPONSES CORRECT	OK?....
DISTRACTORS	OK?....

WRITTEN COMMENTS:

TABLE 3 (cont'd)

PROPORTION ON LAST TEST SELECTING THIS DISTRACTOR:

FIRST	SECOND	THIRD	FOURTH	FIFTH
1.0	1.0	1.0	1.0	1.0

TEST COMMITTEE:

ACCEPT AS IS:	YES....	NO....
MODIFY AS NOTED AND ACCEPT:	YES....	NO....
CATEGORY A.... B....	YES....	NO....
DATE..........		
INITIAL.......		

. . . .

reaches a test committee member (see Figure 1, section (vi)). This does not imply that the reviewer's decision is final, however.

As the reader will see in Table 3, a final check list is provided for the test committee as well. (This listing is performed in the subroutine TEXT.) In this check list, the test committee members may indicate whether or not they will accept the modifications suggested by the reviewer. Provision is also made for the committee member to mark the relative usefulness of an acceptable item, the date, and his initials. The check list for both the reviewer and test committee not only reduces the clerical effort required of users after a retrieval, but also enhances the usefulness of the output.

There can be little doubt that these small, often quite trivial aids, do as much to improve the usefulness of a retrieval as a more fundamental element--creative search strategy. As such they are not to be lightly regarded in designing an information storage retrieval system.

<u>Programming for Iterative Requests: MEDSIRCH-2</u>

As noted before (cf. p. 26) provision for iterative requests has been found as an attractive means of improving the user's satisfaction with his retrievals. The pilot run of MEDSIRCH-1 (which was to be the retrieval of a satisfactory number of multiple choice items for presentation to Internal Medicine's test committee setting the 1969 examination for the Royal College) indicated that provision for iterative requests was necessary in this design also.

Due to the fact that all items are retrieved within a given hierarchy it is possible, for example, that 10 items would be retrieved for a given set of criteria, even though the user had requested only five.

The presence of redundancy has already been discussed (cf. pp. 57-58). In the pilot run the two factors caused MEDSIRCH-1 to retrieve a great many more items than could feasibly be sent to the test committee for consideration. To circumvent this problem the reviewer for Internal Medicine noted the items that were considered most relevant and made a request for those. MEDSIRCH-2 was the program used to search for these modified requests.

To save programming effort the techniques used in coding, file searching, use of direct access media, and the format of input and output used in MEDSIRCH-1 were incorporated into MEDSIRCH-2 (section (v) of Figure 1). As a result the subroutines PDISC, PARMTR, and TEXT are identical in both programs. The subroutine SORT was eliminated, however, since the search was now for specific items, not for characteristics of items with the result that there was no need for retrievals at differing hierarchical levels.

Preparation of search requests. Instead of preparing one card per request as in MEDSIRCH-1 the user had now to prepare two cards per request. The first card, however, was much the same as before. Six specifications are inserted onto this first card: (a) an identification number of the request, (b) the four criteria as before, and (c) the number of items required. Hierarchical level is no longer needed and is therefore omitted.

The second card has its specifications punched as: (a) an identification number of the request, matching the number of the first card; and (b) the bank identification numbers of the items desired. If the user is requesting more than fifteen items in this particular request he

inserts a third crd (or as many as he needs) containing the remaining item identification numbers starting in the first field of five on every such card.

These pairs or groups of cards must be prepared for every set of items meeting the same criteria. A blank card after the last request properly terminates the execution of MEDSIRCH-2. Two requests followed by a blank card are illustrated in Figure 4.

Search technique. The mainline program of MEDSIRCH-2 transfers the tape bank onto disk in the same manner as before. It then reads one group of cards constituting a request. If the identification numbers of this request do no match for these two cards, a diagnostic message indicating the mismatch is written for the user. For example, if a pair had identification numbers of 1 and 10 this message would be given: MISMATCHED ID-PAIRS ON THESE PARAMETER CARDS: 1 AND 10.

The first card of a request is used as information for writing out the chart RESTRICTIONS IMPOSED (Table 3). It also uses the specification "number of items requested" as a means to determine how many item identification numbers to read in the next card(s) of a request.

Storing these item identification numbers in core that have been read on the remaining card(s), the computer then searches through the bank retrieving the items with such identifications. When all items have been found, search is arrested, disk is rewound and another request is processed in the same manner.

From a user's point of view he will see no difference between the output of MEDSIRCH-2 and that of MEDSIRCH-1. The format of the output, the parameter information, checklists, counter, and so forth are iden-

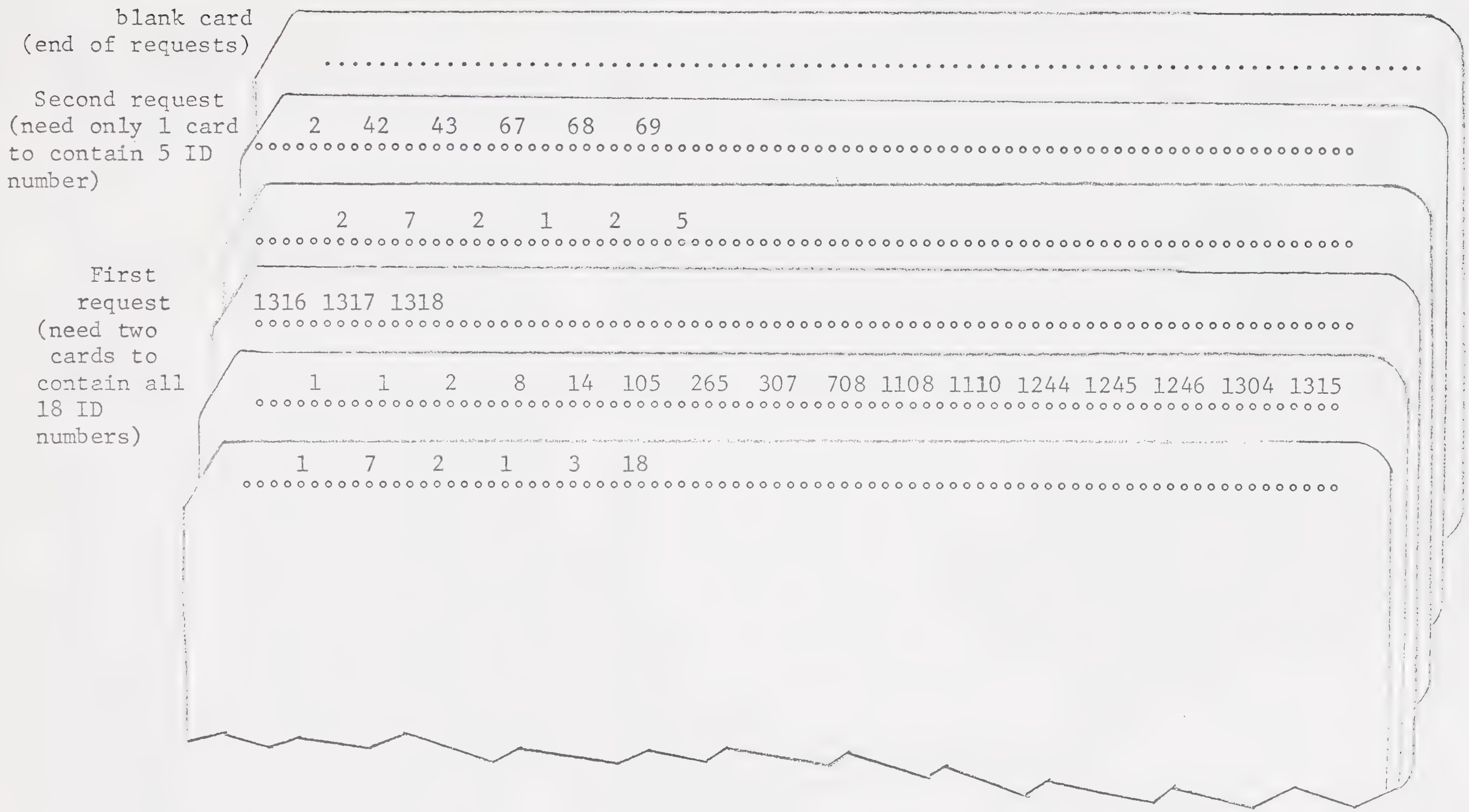

Figure 4. Search Requests for MEDSIRCH-2

tical to MEDSIRCH-1. He will, however, receive only the first message in Table 2 since he has specified in his pair of request cards that all items retrieved belong to one set of criteria.

Programming for Updating Tape Bank: UPDATE

The original bank of items for any information storage and retrieval system must be modified as new material is acquired (section (viii) of Figure 1). In this study, multiple choice items retrieved and used acquire new parameter information, such as revised biserial coefficients, difficulty levels, and so forth. Some of these items may be found to be inadequate and necessitate modifying the text of an item. Other documents may have to be thrown out because advance in medical knowledge has shown the item to be irrelevant. Conversely, new medical knowledge dictates the need for the inclusion of new items. The program UPDATE (Appendix F) was written in order to accomodate all these possibilities.

As with MEDSIRCH-2 attempts were made to reduce programming effort in UPDATE by using some of the techniques of the original program--MEDSIRCH-1.

Preparation of user's requests. Of the five programs used in this study--CHECK, MEDSIRCH-1, MEDSIRCH-2, UTILITY, and UPDATE--the latter requires the most effort on the part of the user, as he must prepare the following cards.

Title card: this first card allows any alphanumeric characters in columns 1-80; whatever the user specifies will be used as a title in his output.

Parameter card: on this card, in fields of five, the user is required to give these specifications--(a) the number of items he wishes

to be deleted from the bank, <u>including</u> those items he is modifying; that is to say, a user must resubmit a whole new item, including its two index cards, if he wishes to make any modification to the text of an item; (b) the number of items having their index cards modified; and (c) the number of items being added as new or modified items.

ID deletion card(s): if the user has specified in the above parameter card that items are to be deleted, he must also submit these "ID deletion card(s)." The user must punch the identification numbers of all items he wishes to be deleted from the bank. As already noted this must include those items that will be modified since the modified items are treated as new additions. Only 16 identification numbers can be punched on a card; if more cards are required numbers are punched on those cards in the same format (fields of five). UPDATE imposes no restrictions on the number of deletions a user wishes to submit.

Item index cards: to modify an item's two index cards in store the user must prepare these two index cards based on the codes in Appendix A. Up to 100 of these pairs of index cards can be submitted in one run of UPDATE.

New or modified items: these are punched in the same manner as before (cf. pp. 46-48). The user must also include an item's two index cards in this submission. UPDATE imposes no limit on the number of new or modified items being added to the bank; the only restriction one must consider is the amount of space remaining in the tape bank.

Updating technique

The procedures available for updating a blocked, sequentially searched data file is limited (cf. p. 14). Therefore, UPDATE uses the technique of rewriting the bank on a new data set--another tape. In practice this is the most advantageous technique, since one has the opportunity of checking the validity of the new tape before destroying the old. This is not possible when revisions are made on the same tape.

While holding in core the identification numbers of all items to be removed, as well as all index cards being inserted as modifications, the mainline program of UPDATE reads through the old tape. Where no changes are made, the old tape is simply copied onto the new tape. If the identification number of an item on the old tape matches any of the identification number of items designated for deletion that item is not rewritten. Similarly if the identification number of an item in the old tape matches that of index cards being used for modification, the old index cards are not written; instead the modified pair of index cards is written. After reaching the end of the old tape, cards containing the items to be added, whether they are new or modified, are read and subjected to the analysis used in the program CHECK (section (vii) of Figure 1). If there are no mistakes they are written on to the new tape. After the final item has been added, the program writes a record with an "E" in column 80 to indicate the end of this new tape.

Diagnostic messages

In order to aid the user in the proper modification of his new bank, UPDATE provides diagnostic messages. A counter is provided at each point where a mistake can be made. If, at the end of the updating, any of these mistakes have been made, a message is given to indicate this:

***NOTE

THIS ATTEMPT TO UPDATE DATA FILE HAS NOT BEEN PROPERLY DONE.

THERE ARE [2] MISTAKES MADE.

REGARD ABOVE MESSAGES TO CORRECT AND RUN THIS PROGRAM AGAIN.

The ABOVE MESSAGES refer to the same diagnostic messages given in Table 1. This reduction in programming effort is made possible since the basic program CHECK is used in UPDATE. There is one contingency, however, that is unique to this program. UPDATE must not only diagnose misplaced parameter cards which accompany a new or modified item (Table 1), but must also diagnose misplaced parameter cards being inserted as modification in themselves. The following message is given for mistakes made in this latter case.

***NOTE

THESE PARAMETER CARDS WHICH ARE BEING USED AS MODIFICATIONS ARE NOT IN ORDER

FIRST PARAMETER CARD READS: CARD [2] ITEM NO. 143

SECOND PARAMETER CARD READS: CARD [1] ITEM NO. 1430

If the user has made no mistakes in using the UPDATE program, messages are given to indicate the satisfactory revisions. Table 4 provides a sample output. The reader will note that in this example items 1, 8, 10, 20, and 25 are modified. As such messages are given to indicate that the old, unmodified items (bearing these identification numbers) have been

TABLE 4

TEST RUN FOR NEW PROGRAM UPDATE

```
THESE ITEMS HAVE BEEN REMOVED :      1
                                     8
                                    10
                                    20
                                    25
                                    31
                                    34
                                    38
                                    40
                                    42

PARAMETER CARDS OF THESE ITEMS ( INDICATED BY THEIR ID NUMBERS )
HAVE BEEN CHANGED:                   2
                                     3
                                     4
                                     5
                                     6

ITEM      1   HAS BEEN ADDED AS A NEW OR MODIFIED ITEM.
ITEM      8   HAS BEEN ADDED AS A NEW OR MODIFIED ITEM.
ITEM     10   HAS BEEN ADDED AS A NEW OR MODIFIED ITEM.
ITEM     20   HAS BEEN ADDED AS A NEW OR MODIFIED ITEM.
ITEM     25   HAS BEEN ADDED AS A NEW OR MODIFIED ITEM.


*** NOTE:
    TO BE SURE TAPE HAS BEEN PROPERLY UPDATED RUN MEDSIRCH2 ASKING FOR
    ITEMS THAT WERE MODIFIED OR ADDED.
```

removed, and the modified items (with the same identification) have been added to the new tape. As a final precaution to the user, he is requested to run the MEDSIRCH-2 program asking for the specific items he has modified in order to verify the updating of his new tape bank (section (v) of Figure 1).

Programming for Efficiency: Job Control Language

The time taken to do a sequential search can be improved by the use of appropriate job control language on systems cards. More detailed descriptions are available elsewhere (IBM, 1967; Hazlett, 1969); however, the IBM 360/67 system's parameters and subparameters specific to this study are now described.

Data Control Block (DCB). This parameter was used for specifying record formats (RECFM), blocksizes (BLKSIZE), logical record lengths (LRECL), and densities (DEN). Blocking tends to increase the reading efficiency of one's data set by approximately the order of the blocking factor. That is to say, if one has a blocking factor of 10, then the efficiency is approximately 10 times as great as a data set with no blocking. There is a point, however, at which efficiency begins to drop off; this is determined by the total core storage being used, and is peculiar to each program. Blocking was used in this design for both the tapes and for the direct access medium of disks.

Both the programs UTILITY and UPDATE (sections (iv) and (viii) of Figure 1) essentially write all records onto tape in card image. As such, the records are fixed length and are under format control. Card five in the utility program (Appendix C) specifies the exact DCB parameter

for this study. The explanation of each of the subparameters is as follows.

Since the tape is stacked in card image, RECFM must be fixed blocked (FB) since records are indeed blocked and are the same fixed length as a card (80 columns). This in turn determines that the maximum READ/WRITE statement used for the tape must be no longer than 80 columns; thus LRECL is 80. While tapes may have any blocksize, efficiency was not noticeably increased after a blocking factor of 90; therefore, blocksize was set at 7200 bytes. Since all tapes used on the IBM 360/67 should be nine-track the only allowable density is 2.

Before any searches are begun, the entire bank is rewritten onto disk without format in order to increase the economy of processing searches. The lack of format control presents some difficulty, however, in efficient blocking. Whereas writing _with_ format control LRECL was found by counting the maximum number of _characters_ (columns) in a READ/WRITE statement, reading or writing _without_ format control LRECL is the maximum number of _variables_ multiplied by four (the number of bytes). In the text of an item, a record without format control contains 19 x 4 = 76 bytes. The first parameter card contains 40 x 4 = 160 bytes and the second parameter cards have 19 x 4 = 76 bytes. Thus the records which were originally of fixed length on tape are now, on disk, of variable length. (Since this is generally the case, only variable length records may be written without format control.)

Since LRECL is four more bytes than the maximum record length one would ordinarily specify this as 164. With an optimal blocking factor of 90 the BLKSIZE would be (90 x 164) + 4 = 14964. (For further explanations regarding calculating procedures see the literature (IBM, 1967,

pp. 44-46; Hazlett, 1969, pp. 15-19). This procedure is unacceptable since maximum BLKSIZE for disk is 7294 bytes.

To circumvent this difficulty all programs--MEDSIRCH-1, MEDSIRCH-2, and UPDATE--write the parameter information accompanying each item as three, not two, records. The first and second record are 19 variables long, hence 76 bytes; the third has 17 variables, therefore, 17 x 4 = 68 bytes.

Since the maximum READ/WRITE statement without format control is now 76, LRECL is 80. Using a blocking factor of 90 one obtains a BLKSIZE of 7204, which is under the maximum size of 7294. Thus one has kept the optimal blocking factor of 90 while still keeping within the limitation of blocksize for disk.

Separation Subparameter (SEP). One further feature was incorporated to optimize the efficiency of the sequentially searched file in MEDSIRCH-1, that of providing separate access arms to each of the data sets on disk. The reader will recall that four different data sets on disk were used: (a) data set one for the main file, and (b) data sets two, three, and four containing items retrieved at various hierarchical levels. The access arm containing the READ/WRITE head in the computer must move through a physical distance when reading or writing on different disks. The SEP subparameter decreases access time to data sets by providing separate access arms to each data set. Though this separation device may be ignored by the operating system if an insufficient number of access-arms are available this feature does optimize execution time whenever honored.

<u>Space parameter (SPACE)</u>. In order to insure that adequate space was allotted to the running of these programs, a space parameter was also included. All contingencies encountered could be handled in the specification SPACE=(TRK,(10,10). In other words space was reserved for 10 tracks, with option of allowing up to 10 more tracks when previously allocated space was exhausted.

Table 5 contains a listing of systems cards used for the two MEDSIRCH programs as well as UPDATE. The other specifications in this Table that have not been dealt with in detail here are required when using any tapes or disks on the IBM 360/67 and as such need no specific description.

In this chapter the reader has been given the reasons for the choice of this particular design in information retrieval, the techniques used in indexing and searching, the preparation user requests and provision of iterative searches, means for reducing user errors and provision for diagnostic messages, and finally suggestions for optimizing the retrieval efficiency in terms of time and programming effort. At no point has there been a critical evaluation of this design. Since no design can be accepted without appraisal one must look at this design's relative worth as an information storage and retrieval system. Chapter IV deals with this evaluation and the implications stemming from such an appraisal.

TABLE 5

SYSTEM CARDS FOR CREATION OF DATA SETS

Program	System Cards	
MEDSIRCH-1	//GO.FT01F001 DD UNIT=SYSDA,SPACE=(TRK,(10,10)),	C
	// DCB=(RECFM=VB,BLKSIZE=7204,LRECL=80),	C
	// DISP=(NEW,DELETE)	
	//GO.FT02F001 DD UNIT=(SYSDA,SEP=(FT01F001)),	C
	// SPACE=(TRK,(10,10)),	C
	// DCB=(RECFM=VB,BLKSIZE=7204,LRECL=80),	C
	// DISP=(NEW,DELETE)	
	//GO.FT03F001 DD UNIT=(SYSDA,SEP=(FT01F001,FT02F001)),	C
	// SPACE=(TRK,(10,10)),	C
	// DCB=(RECFM=VB,BLKSIZE=7204,LRECL=80),	C
	// DISP=(NEW,DELETE)	
	//GO.FT04F001 DD UNIT=(SYSDA,SEP=(FT01F001,FT02F001,FT03F001)),	C
	// SPACE=(TRK,(10,10)),	C
	// DCB=(RECFM=VB,BLKSIZE=7204,LRECL=80),	C
	// DISP=(NEW, DELETE)	
	//GO.FT08F001 DD DSNAME=MEDS,UNIT=SYSTP,	C
	// DCB=(RECFM=FB,BLKSIZE-7200,LRECL=80,DEN=2),	C
	// VOLUME=SER=T00525,DISP=(OLD,KEEP)	

TABLE 5 (cont'd)

MEDSIRCH-2	//GO.FT08F001 DD DSNAME=MEDS.UNIT=SYSTP,	C
	// DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80,DEN=2),	C
	// VOLUME=SER=T00525,DISP=(OLD,KEEP)	
	//GO.FT01F001 DD UNIT=SYSDA,SPACE=(TRK,(10,10)),	C
	// DCB=(RECFM=VB,BLKSIZE=7204,LRECL=80),	C
	// DISP=(NEW,DELETE)	
UPDATE	//GO.FT08F001 DD DSNAME=MEDS,UNIT=SYSTP,	C
	// DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80,DEN=2),	C
	// VOLUME=SER=T00525,DISP=(OLD, KEEP)	
	//GO.FT01F001 DD DSNAME=UPDA,UNIT=SYSTP,	C
	// DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80,DEN=2),	C
	// VOLUME=SER=T00725,DISP=(NEW,KEEP)	

CHAPTER FOUR

SUMMARY, EVALUATION, AND RECOMMENDATIONS

Summary

In order to focus one's attention to those areas of this study that must be evaluated, a restatement of the important features of this design follows. The purpose was to design and implement an efficient, broadly useful technique whereby educators could reduce the clerical work involved in keeping and selecting multiple choice items for examination purposes. Of the three basic data files--sequential, random, and list--the first was chosen in order to best meet the needs and demands of users and of the capabilities of available computer hardware. Manual indexing was also chosen as the most promising means of retrieving items, even though the process involved inefficiency and the cost of human effort.

A detailed description was given as to the nature of the indexes and items, the format for their specification, and means for storing them as a bank on magnetic tape, to be searched with the program MEDSIRCH-1. The search technique, the weighting of retrievals, and nature of output was also explained. In order to allow a user to receive the most pertinent records, MEDSIRCH-2 was provided for modified requests.

Since no design in the field of information is complete without provisions for modifying the store of information, the program UPDATE was intended to serve this function, providing means for additions, deletions, and modifications.

Throughout the entire design, steps were taken to reduce human effort: (*a*) user's requests required minimal specifications in the most simple of formats; (*b*) diagnostic messages were provided in the output, indicating as clearly as possible the exact type of error made by a user; (*c*) retrievals were titled in such a manner as to indicate their degree of relevance, other aids such as check lists and perforated pages increasing the utility of retrieval, and (*d*) programming overlap, while accomplishing different tasks, reduced programming effort and cost.

Furthermore, the attempts made to anticipate future needs, such as indexes for audio-visual material, graduate or undergraduate, local or national and speciality number will, presumably, reduce future modifications to the indexing code of Appendix A.

Last but not least, job control language was found to be a means for reducing execution time of the programs used in this design. Parameters and subparameters, such as data control blocks, record formats, blocksizes, separation devices, all contributed to increased efficiency. Their exact specification, however, was determined by the use of another means in attaining efficiency, that of direct access disk. Disks not only allowed the direct transfer of binary information but reduced the time for REWIND statements.

This study has provided no evidence for estimating the cost, the efficiency or the degree of recall and relevance and its generality or contribution to the field of education and information science. This, of course, is of importance and will now be discussed.

Evaluation

The evaluation of this design will be carried out under three main headings: (a) usefulness, (b) cost, (c) generality. Unfortunately a rigorous evaluation of such a design is not possible unless other systems can be used to estimate its relative worth. This study has made no attempt to implement other designs for this purpose. In many instances, therefore, evaluative claims are more subjective than objective, and as such must be considered in that light. The literature (Jahoda, 1968; Spring, 1967; Vickery, 1965) indicates that most analyses of information search procedures, and measurements of retrieval performances, are saddled with similar difficulties. This report, however, will give recall, relevance and cost figures in order that further studies may use this design for comparison.

Usefulness

Ernst (1965) provides two mathematical models with which to evaluate the usefulness and pertinency of retrievals. The first is the recall rate--the fraction of the relevant material which is retrieved; the second is the precision ratio--the fraction of retrieved material which is relevant. Authors (Shoffner, 1968; Tritschler, 1968; Cooper, 1968; Bennett, 1966; Savage, 1964) criticize such models since relevance is difficult to determine. The problem lies in the assumption that there is such a person as *the* user. Users as humans are different and have different purposes. Relevance, therefore, is not easily dichotomized.

While these authors have pointed out the weaknesses in such an analytical model they have little to offer in terms of improving it.

In order to standardize the interpretation of recall and precision rate this investigator has chosen to define relevance in precise, measurable terms for this design: a relevant retrieval is one that meets the four restrictions submitted by the user. Though there is some loss of information in limiting the meaning to this definition, there is reason to assume that recall and precision rates will not be overly, if at all, inflated. While it is true that some items meeting the four restrictions submitted by the user were not regarded as useful by, for example, the reviewer, there were items which were regarded as acceptable, though they were retrieved at a lower hierarchial level than those meeting all four criteria. (In the pilot run of this retrieval system, about one-half of the items out of a total of 201 that were presented to the test committee for final screening were of this nature.)

Using this definition of "relevance", calculated recall and precision rates would tend to be too great whenever a user rejected "relevant" material. Conversely, the calculated rates would be too low whenever a user accepted "nonrelevant" material. By virtue of the fact that both conditions are present in most, if not all, retrievals, it will be assumed that their influence on the recall and precision rates is nullified. Provided this assumption is valid, Ernst's analytical model is an appropriate one to use for this design. The following rates are reported in the context of this limitation.

For MEDSIRCH-1 the recall rate was found to be 100%, that is, all items in the bank that could meet the user's specifications were retrieved. Verification of this was done on the card sorter. This excellent recall figure is a direct result of using manual instead of automatic indexing. Precision rate was considerably poorer. Six hundred items were retrieved from the bank which has only 425 items; this redundancy was a direct result of not incorporating the sixth specification on the user's request --hierarchical level. Of these 600 items 201 were selected by the reviewer. Thus the precision rate was about 33%. If the user had used the hierarchical level specificaiton, it is reasonable to assume that no more than 425 items, at most, would have been retrieved. In such a case the precision rate would have increased to 47% at least.

Due to the fact that MEDSIRCH-2 retrieves items according to identification numbers, it is not surprising that the recall and precision rates in the pilot run were 100%. In practice, therefore, the program is only used as a means for grouping all pertinent items retrieved in MEDSIRCH-1, and then printing them out in a convenient format.

The precision rate of MEDSIRCH-1, even estimated at 47%, is unacceptable. If an entire bank is retrieved one might as well have the items in a manual file; not only would the cost be reduced but time would be saved. However, the limited size of the bank and the nature of the items stored therein caused much of this difficulty. For example, while there were 20 items with the combined specifications of 2-1-2-1 there were no items meeting even the first restriction of 12-1-1-1, much less all four restrictions. Almost one quarter of all the items were concentrated in three subspecialities; and even within each of these three

subspecialities there was not a good distribution of items throughout the hierarchy of criteria. For example, though the above specification of 2-1-2-1 had 20 items, there were only two items with 2-1-3-1, and none with 2-2-3-3. It was not surprising, therefore, that a series of requests having little in common with what was available in the bank caused the precision rate to be low.

To circumvent this problem, future users must have at their disposal freqency tables indicating the distribution of criteria. This, fortunately is already available. Reading a tape stacked with only the parameter cards of all the items a cross-classification program using simultaneous subdivisions will provide such information (see documentation by Flathman, 1968). With this a priori knowledge, the precision rate may reach 100%.

Another factor detracted from the usefulness of this program. It was decided by the test committee using the pilot retrieval that the specification "core level" was not a useful criterion for retrieval. Since an examination was intended to cover all 23 subspecialities at three taxonomic levels while still using a reasonably short examination, it was felt the Department of Internal Medicine could not afford the luxury of testing anything but essential material. (Whether or not all specialities will feel the same way remains to be seen.) In any case, for future retrievals in Internal Medicine search criteria will be reduced to three specifications.

Two of Dammers' (1968) objectives in a retrieval system are simplicity of input and a directly useable output. Since the MEDSIRCH programs use a consistent format (fields of five) and require minimal preparation

of requests, a claim can be made that the first objective of Dammers is met. Since an item is printed in a manner that can be directly typed into an examination (along with the other user aids that were noted, cf. pp. 58-65), objective of usable output is at least met to a fair degree. However, when one considers the precision rate of the first retrieval for MEDSIRCH-1 it must be said that the output was not "directly" useable.

Cost

Since monetary cost is a localized estimate, only cost in terms of time will be discussed. Table 6 provides the estimated cost per item entailed in the pilot run of this information storage and retrieval system. One will notice that the cost of preparing items for storage--typing and keypunching--is twice the amount entailed in a manual file. The time spent selecting items for MEDISRCH-1 for requests to MEDSIRCH-2 can be very great whenever 600 items must be manually searched. The work entailed by the reviewer would be common to any retrieval system--manual or machine.

The time taken to retrieve an item is probably faster than that required for a manual search even if a person was not sequentially searching a file as the computer did. Blocking tape and disks is directly responsible for the small amount of time spent in reading and writing; in fact this input/output process entailed only 5% of the execution time. However, with today's high speed computers such as the IBM 360/67 one cannot help but wonder how much time could be saved with the use of random direct access or list files.

TABLE 6

COST PER ITEM

(in terms of minute per item)

	Minutes per item
Man	
Typing item	7.0
Coding	
New Item	3.0
Old Item	6.0
Keypunching item and its index card	7.0
Selecting relevant items from	
MEDSIRCH-1 retrieval for MEDSIRCH-2	5.0
Reviewer's working in checking content, spelling, etc.	5.0

Computer	Minute per item retrieved or updated	
Program	Reading/ Writing	Total Execution
MEDSIRCH-1	0.003	0.048
MEDSIRCH-2	0.002	0.040
UPDATE	0.0017	0.034

Generality

One objective of this design was for the broad application of such a system to other educational areas. This objective was at least partially fulfilled. Provided the multiple choice questions are punched according to format requirements (cf. pp. 46-48), all programs will handle the text of items regardless of the educational area. The indexing code in Appendix A is not so generalizable. Specifications such as subspeciality, second area of subspecialty, source, graduate or undergraduate, local or national, and speciality number are presently converted by means of data statements in the two MEDSIRCH programs, refer to the exact needs of Internal Medicine and could not be used for grade twelve examinations, for example. Even for other medical specialities the subspecialty codes are not applicable. There are means for adjusting the MEDSIRCH programs to fit most needs, however.

To modify the programs for use by other medical departments involves the least work. For the specialty code one would have to take out the present data statements and insert ones containing references to the particular department making the modification. Since two columns are provided for these specifications on the first parameter card one could have up to 99 subspecialties.

To adapt the program for use in the school system would require changing more than one data statement, but still involves little effort relative to the time required in designing a new system. For example, the two codes for subspecialty could be changed to subject area (Physics, Mathematics, Social Studies, etc.) and section (levers, polynomials, geography, etc.); province could be changed to grade, taxonomic level

modified to include the hierarchical levels suggested by Bloom, and source to cognitive or affective domains. These are, of course, only suggestions; the precise nature of codes would be determined by consulting the intended users. One would not be restricted to keeping exactly the same format for these codes, nor would he be required to use the same order of criteria or even the same criteria. If any of these modifications were desired, however, more programming effort would be involved. Regardless of the number of changes made this system's file organization, search technique, and optimization procedures can provide a means for reducing considerably the time required for creating an information storage and retrieval system for anyone using multiple choice questions.

Finally, to use this design's programs at other installations, Appendix G provides a list of hardware specifications that must be met. Included in this appendix are modifications that may be made in order to tailor the design to smaller or less sophisticated computers.

Recommendations

Suggestions have already been made as to how the retrieval process of this design may be inproved, namely, the use of the hierarchical level specification, the use of cross-classification tables, and the removal of the restriction core level. Whenever the bank acquires an equal frequency distribution of items throughout all possible levels it would be desireable to limit the number of retrievals to the exact number of items the user requests since, at present, all items are retrieved at a given hierarchical level. Given more items than needed by the present

search technique, a ranking procedure based on the size of correlation coefficients (between a user's request and an item's remaining indexes) could be incorporated. For example, one could use further specifications such as second area of subspecialty, language, number of times used, last year question used, difficulty levels, and biserial coefficients as binary elements in each of two vectors--a user's request and an item's remaining information. The degree of angle between these two vectors would then be calculated for all items. Ordering items according to their relative size of angles, retrieval would take place until the exact number of items requested was met. Since new items would have missing information for most of these specifications one might want to retrieve these first and then retrieve whatever else is needed according to this ranking procedure.

As the evaluation of this program is very subjective, further programming should be done with the bank, using random direct access and list file organizations, in order to determine which is the best design in terms of time, cost, and effort. Findings may support Climenson's (1967) claim that combinations of these file systems provide optimal means for retrieval.

Despite the number of recommendations one can make it is the opinion of this writer that most, if not all, attempts to improve or evaluate present information storage and retrieval designs is analogous to a scheme for improving the destruction of an iceberg by blasting it from the top. When the full truth about the bottom of the iceberg--that is retrieval design--is fully realized, the time spent in estimating

the relative worth of this design will be seen as poorly spent. Until such a time, simple systems such as the one described in this study seem to offer a reasonable means for storing and retrieving multiple choice questions.

SELECTED REFERENCES

Abelson, P. H. Custodians of knowledge. _Science_, 1968, _159_, 1.

Abramson, N. _Information theory and coding_. New York: McGraw-Hill, 1963.

Altmann, B. A natural language storage and retrieval (ABC) method: its rational, operation, and further development program. _Journal of Chemical Documentations_, _6_(3), 154-157.

Baruch, J. J. Information system applications. In C. A. Cuadra (Ed.), _Annual review of information science and technology_. Vol. 1. New York: John Wiley & Sons, 1966.

Belz, K. A customer information system for utilities. _Burotechnik & Automation_, 1966, _12_, 527-730. (_Computing Reviews_, 1967, _8_(5), 438-439.)

Bennett, J. L. Review of D. R. Swason, On indexing depth and retrieval effectiveness. _Computing Reviews_, 1966, _7_(1), 49.

Benson, R. A. _Information storage and retrieval methods for the astronautics library_. RD-1 No. 111-9047-911, General Dynamics/Astronautics, San Diego. Cited by D. V. Blank & E. A. Farley, Library automation. In C. A. Cuadra (Ed.), _Annual review of information science and technology_. Vol. 1. New York: John Wiley & Sons, 1966.

Black, D. V. & Farley, E. A. Library automation. In C. A. Cuadra (Ed.), _Annual review of information science and technology_. Vol. 1. New York: John Wiley & Sons, 1966.

Bleier, R. E., & Vorhaus, A. H. File organization in the SDC time-shared data management system (TDMS), _Proceedings IFIP Congress 1968_, Amsterdam: North-Holland, 1968. (_Computing Reviews_, 1969, _10_(1), 23-24.)

Bloom, B. S. Taxonomy of educational objectives. Handbook 1: Cognitive domain. New York: David McKay, 1956.

Borko, H. Studies on the reliability and validity of factor-analytically derived classification categories. In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.), Statistical association methods for mechanized documentation. Washington, D. C.: United States of America Government Printing Office, 1964.

Bryant, E. C. A status report on research in information retrieval. In J. M. Hankes (Ed.), Management information systems and the information specialists. Layfayette: Purdue University, 1966.

Clemenson, W. D. File organization and search techniques. In C. A. Cuadra (Ed.), Annual review of information science and technology. Vol. 1. New York: John Wiley & Sons, 1966.

Cooley, S. E. G. Medical bibliography and MEDLARS. Plastic and Reconstructive Surgery, 1967, 40, 113-116.

Cooper, W. S. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems, American Documentation, 1968, 19(1), 30-41. (Computing Reviews, 1969, 10(4), 192.)

Cuada, C. A. Annual review of information science and technology. Vol. 1. New York: John Wiley & Sons, 1966.

Cuada, C. A. Annual review of information science and technology. Vol 2. New York: John Wiley & Sons, 1967.

Cuada, C. A. Annual review in information science and technology. Vol. 3. New York: John Wiley & Sons, 1968.

Dammers, H. F. Integrated information processing and the case for a national network, Information Storage and Retrieval, 1968, 4(2), 113-131.

Dodd, G. G. Elements of data management systems. Computing Surveys, 1969, 1(2), 117-133.

Doyle, L. G. Some compromises between word grouping and document groups. In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.). Statistical association methods for mechanized documentation. Washington, D. C.: United States of America Government Printing Office, 1964.

Doyle, L. B. Review of G. Salton, Automatic information organization and retrieval. Computing Reviews, 1969, 10(6), 271-272.

Edmundson, H. P. A correlation coefficient for attributes or events. In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.). Statistical association methods for mechanized documentation. Washington, D. C.: United States of America Government Printing Office, 1964.

Ernst, M. L. Evaluation of performance of large information retrieval systems. In J. Spiegel & D. Walker (Eds.), 2nd. Congress on the Information System Sciences. Washington, D. C.: Spartan Books, 1965. (Computing Reviews, 1966, 7(1), 43-44.)

Flanagan, D. Information. San Fransisco: W. H. Freeman, 1966.

Flathman, D. P. Cross classification with Sub-division (NONPØ9). Unpublished manuscript, University of Alberta, 1968.

Garfield, E. Can citation indexing be automated? In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.). Statistical association methods for mechanized documentation. Washington, D. C.: United States of America Government Printing Office, 1964.

Greenberger, M. The use of computers. In D. Flanagan (Ed.), Information. San Francisco: W. H. Freeman, 1966.

Gove, P. B. Problems in defining. In J. H. Shera, A. Kent, & J. W. Perry (Eds.), Information systems in documentation. Vol. 2. New York: Interscience Publishers, 1957.

Hammond, W. Statistical association methods for simultaneous searching of multiple document collections. In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.), Statistical association methods for mechanized documentation. Washington, D. C.: United States of America Government Printing Office, 1964.

Hazlett, C. B. Creation of Data Sets on the IBM 360/67. Unpublished manuscript, University of Alberta, 1969.

Hubbard, V. P. & Clemans, W. V. Multiple-choice examinations in medicine. Philadelphia: Lea & Febiger, 1961.

International Business Machines. Introduction to IBM system/360 direct access storage devices and organization methods. White Plains: IBM Technical Publications Department, 1966.

International Business Machines. IBM system/360 operating system fortran IV(H) programmer's guide. White Plains: IBM Technical Publications Department, 1967.

Jahoda, G. Review of G. Salton, Information storage and retrieval ISR-13: reports on evaluation procedures and results 1965-1967. Computing Reviews, 1968, 9(7), 385.

Jones, K. S. The use of automatically-obtained keyword classifications for information retrieval: Final report. Cambridge: Cambridge Language Research Unit, 1969.

Kent, A. Computers and biomedical information storage and retrieval. Journal of American Medical Association, 1966, 196, 927-932.

King, D. W. Design and evaluation of information systems. In C. A. Cuadra (Ed.), Annual review of information science and technology, Vol. 2. New York: John Wiley & Sons, 1968).

Kuhns, J. L. The continuum of coefficients of associations. In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.), Statistical association methods for mechanized docuementation. Washington, D. C.: United States of America Government Printing Office, 1964.

Kurtzke, J. F. Caveat computator. Methods of Information in Medicine, 1968, 6, 127-129. Cited by R. P. Levy & M. R. Cammarn, Information systems applications in medicine. In C.A. Cuada (Ed.), Annual review of information science and technology. Vol. 3. New York: John Wiley & Sons, 1968.

Licklider, J. C. R. Man-computer interaction in information systems. In M. Rubinoff (Ed.), Toward a national information system. Washington, D. C.: Spartan Books, 1965. (Computing Reviews, 1966, 7(4), 318-319.)

Lipetz, B. Information storage and retrieval. In D. Flanagan (Ed.), Information. San Francisco: W. H. Freeman, 1966.

Mathews, F. W. & Thomson, L. Weighted term search: a computer program for an invented coordinate index on magnetic tape. Journal of Chemcial Documents, 1967, 7(1), 49-56.

McCarthy, J. Information. In D. Flanagan (Ed.), Information. San Francisco: W. H. Freeman, 1966.

Minker, J. Review of G. Salton, Search and retrieval experiments in real-time information retrieval. Computing Reviews, 1969, 10(4), 193-194.

Parker, E. B. The user's place in an information system. American Documentations, 1966, 17(1), 26-27. (Computing Reviews, 1966, 7(5), 400.)

Rubinoff, M., Bergman, S., Cautin, H., & Rapp, F. EASY ENGLISH, a language for information retrieval through a remote typewriter consol. Communications of the ACM, 1968, 11(10), 693-696.

Sage, C. R. Adaptive information dissemination. American Documentations, 1965, 16(3), 185-200. (Computing Reviews, 1966, 7(3), 240.)

Salton, G. Automatic information organization and retrieval. New York: McGraw-Hill, 1968.

Saltzberg, B. What is information theory? In W. S. Fields & W. Abbot (Eds.), Information storage and neural control. Springfield: Charles C. Thomas, 1963.

Savage, T. R. The evaluation of automatic indexing and classification. In D. Flanagan (Ed.), Information. San Francisco: W. H. Freeman, 1966.

Schoffner, R. M. Organization, mainenance and search of machine files. In C. A. Cuadra (Ed.), Annual review of information science and technology. Vol. 3. New York: John Wiley & Sons, 1968.

Soergel, D. Some remarks on information languages, their analysis and comparison. Information Storage and Retrieval, 1967, 3(4), 219-291. (Computing Reviews), 1968, 9(9), 536-537.)

Sprague, R. H., Fr. A comparison of systems for selectively dissimenating information. Bloomington: Indiana University, 1965. (Computing Reviews, 1966, 7(3), 239.)

Spring, W. C., Jr. Applications in Medicine. In C. A. Cuadra (Ed.), Annual review of information science and technology. Vol. 2. New York: John Wiley & Sons, 1967.)

Stevens, M. W. Automatic indexing using cited titles. In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.), Statistical association methods for mechanized documentation. Washington, D. C.: United States of America Government Printing Office, 1964.

Stewart, B. The serials mechanization of the Texas A & M University library. Unpublished manuscript, Austin, 1966. Cited by B. A. Markuson, Automation in libraries and information centers. In C. A. Cuadra (Ed.), Annual review of information science and technology. Vol. 2. New York: John Wiley & Sons, 1967.

Stiles, H. E. Automatic indexing and the association factor. In S. M. Newman (Ed.), Information systems compatibility. Washington, D. C.: Spartan Books, 1965.

Taylor, R. S. Progessional aspects of information science and technology. In C. A. Cuadra (Ed.), Annual review of information science and technology. Nol. 1. New York: John Wiley & Sons, 1966.

Tinker, J. F. Imprecision in meaning measured by inconsistency of indexing. American Documentations, 1966, 17(2), 96-102. (Computing Reviews, 1967, 8(2), 119.)

Tritschler, R. J. Review of J. W. Tukey, A statistician's comment. Computing Reviews, 1968, 9(9), 541.

Vickery, B. C. On retrieval system theory. (2nd ed.) Washington, D. C.: Butterworth, 1965.

Wallace, E. M. Rank order patterns of common words as discriminators of subject content in scientific and technical prose. In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.), Statistical association methods for mechanized documentation. Washington, D. C.: United States of America Government Printing Office, 1964.

Williams, J. H. Results classifying documents with multiple discriminant functions. In M. A. Stevens, V. E. Guiliano, & L. B. Heilprin (Eds.), Statistical association methods for mechanized documentation. Washington, D. C.: United States of America Government Printing Office, 1964.

Wittman, B. A. & Ingerman, P. Z. A threshold selection language. Proceedings of the ACM 22nd National Conference. Washington, D. C.: Thompson Book Co., 1967. (Computing Reviews, 1968, 9(8), 446.)

A P P E N D I X   A

Indexes for Items in Internal Medicine

A.1: Preparation of Parameter Cards

Please note, to correctly code each item remember these three points.

1. TWO parameter cards are needed for each item. Even if no item information is available for the second card it is still necessary to punch the specialty number 01 (Internal Medicine) in col. 74-75 and card number "2" in col. 72 and item ID in col. 77-80 and place this otherwise blank card with the first parameter card.

2. It is imperative that information for the first four variables [(a) subspecialty, (b) type of question, (c) taxonomic level, and (d) core level] be given. If any one of the first four is omitted the item will never be selected.

3. Use right justification throughout.

A.2: Card 1

Column

1-2 Area of Subspecialty

Punch "1"	Allergy, Immunology, Serology	(ALL)
"2"	Cardiovascular	(CVS)
"3"	Collagen Diseases	(COL)
"4"	Dermatology	(DERM)
"5"	Chemical of Physical Agents	(PHYSCHEM)
"6"	Endocrinology and Metabolism	(END MET)
"7"	Gastrointestinal (incl. Liver & Pancreas)	(GI)
"8"	Hematology	(HEMAT)
"9"	Infectious Diseases	(INF)
"10"	Musculoskeletal	(SKEL)
"11"	Neurology	(NEUR)
"12"	Psychological Medicine	(PSYC)
"13"	Pulmonary	(PULM)
"14"	Renal	(REN)
"15"	Therapeutics	(THER)
"16"	Anatomy	(ANAT)
"17"	Biochemistry	(BIOC)
"18"	Genetics	(GEN)
"19"	Laboratory Medicine	(LABMED)
"20"	Microbiology	(MICROB)
"21"	Pathology	(PATH)
"22"	Pharmacology	(PHARM)
"23"	Physiology	(PHYSIOL)

Column

3	Type of Question	
	Punch "1" Single Answer	(SING ANS)
	"2" Multiple Answer	(MULT ANS)
4	Taxonomic Level	
	Punch "1" Factual	(FACT)
	"2" Comprehension	(COMP)
	"3" Problem Solving	(PROB)
5	Core Level	
	Punch "1" Essential	(ESS.)
	"2" More Important	(IMP.)
	"3" More Unimportant	(UIMP)
6-7	Second Area of Subspecialty (re. Col. 1-2)	
8	Source	
	Punch "1" American Board of Internal Medicine	(AMIB)
	"2" National Board of Medical Education	(NBME)
	"3" Canada	(CAN)
	"4" United Kingdom	(UK)
	"5" Other	(OTH)
9-10	Province	
	Punch "1" Alberta	(ALTA)
	"2" British Columbia	(B.C.)
	"3" Dalhousie	(DALH)
	"4" Lavalle	(LAVL)
	"5" McGill	(MCG)
	"6" McMaster	(MCM)
	"7" Manitoba	(MAN)
	"8" Montreal	(MTRL)
	"9" Ottawa	(OTT)
	"10" Queens	(QN)
	"11" Saskatchewan	(SASK)
	"12" Sherbrooke	(SHRB)
	"13" Toronto	(TOR)
	"14" Western Ontario	(UWO)
	"15" Calgary	(CALG)
	"16" Memorial	(MMRL)

Column

11 Audio-Visual

Punch "1" Line
"2" Photo
"3" Color
"4" Slide
"5" Movie
"6" Video

12-14 Audio-Visual I.D. location

15-19 Correct Response Alternative(s)

"1" in col. 15 choice 1 is correct
"1" in col. 16 choice 2 is correct
"1" in col. 17 choice 3 is correct
"1" in col. 18 choice 4 is correct
"1" in col. 19 choice 5 is correct

20 Language

"1" available in both English and French
"2" available in English only
"3" available in French only

21 Number of times used

(If zero there should be no more entries on this or the next parameter card except for specialty, card and item ID #)

22-23 Last year question used

24-25 Number of question on last exam

26 "1" Graduate exam
"2" Undergraduate exam

27 "1" National exam
"2" Local exam

28-30 ID of exam

31-34 Number of examinees on last exam

Use col. 35-42 only for single-answer type of questions. Use col. 43-66 of parameter card 1 and col. 1-24 of paramter card 2 only for multiple-answer type of questions.

column	
35-36	"p" for last recorded testing year (single answer-type of question)
37-38	"p" for second last recorded testing year (single-answer type of question)
39-40	r_{bis} for last recorded testing year (single-answer type of question)
41-42	r_{bis} for second last recorded testing year (single-answer type of question)
	Multiple-Answer Type of Question: "P" for last recorded testing year.
43-44	First choice
45-46	Second choice
47-48	Third choice
49-50	Fourth choice
51-52	Fifth choice
53-54	Total item
	Multiple-Answer Type of Question: r_{bis} for last testing year
55-56	First choice
57-58	Second Choice
59-60	Third Choice
60-62	Fourth choice
63-64	Fifth choice
65-66	Total Item
74-75	Specialty Punch "01" for Internal Medicine
76	Punch "1"
77-80	Item ID (must be identical to the number in col. 74-77 of cards carrying the test of the item).

A.3: Card 2

Column	
	Multiple-Answer Type of Question: "P" for second last recorded testing year.
1-2	First choice
3-4	Second choice
5-6	Third choice
7-8	Fourth choice
9-10	Fifth choice
11-12	Total item
	Multiple-Answer Type of Question: r_{bis} for second last recorded testing year.
13-14	First choice
15-16	Second choice
17-18	Third choice
19-20	Fourth choice
21-22	Fifth choice
23-24	Total Item
	Proportion on last test selecting these choice
25-26	First choice
27-28	Second choice
29-30	Third choice
31-32	Fourth choice
33-34	Fifth choice
74-75	Specialty, Punch "01" for Internal Medicine
76	Punch "2"
77-80	Item ID (must be identical to ID in col. 77-80 of first card)

A P P E N D I X   B

Program MEDSIRCH-1

```
C
C     MEDSIRCH-1   DIVISION OF EDUCATIONAL RESEARCH SERVICES
C                               UNIVERSITY OF ALBERTA
C     ..................................................................
C
C                  SEARCH FOR MEDICAL EXAMINATION QUESTIONS
C                  ROYAL COLLEGE OF PHYSICIANS AND SURGEONS
C                      DEPARTMENT OF INTERNAL MEDICINE
C
C     PROGRAMMER:  C.B.HAZLETT
C
C     PURPOSE:
C        1. READS MULTIPLE CHOICE ITEMS FROM TAPE AND SELECTS
C           THOSE MEETING THE FIRST FOUR RESTRICTIONS USER REQUIRES.
C        2. IF USER REQUIRES MORE ITEMS THAN NUMBER MEETING FOUR
C           RESTRICTIONS,THOSE MEETING THE FIRST THREE (THEN TWO,
C           THEN ONE) ARE ALSO GIVEN, IF REQUESTED.
C
C     CARD INPUT:
C        1. PARAMETER CARD
C             A. USE RIGHT JUSTIFICATION.
C             B. ONLY ONE CARD NECESSARY FOR EACH SET OF RESTRICTIONS.
C        2. LAST CARD: BLANK CARD
C
C     DESCRIPTION OF PARAMETER CARD (5I5)
C        MED    - AREA OF SUBSPECIALITY
C               - IF MED= 1  ALLERGY,IMMUNOLOGY,SEROLOGY
C                       = 2  CARDIOVASCULAR
C                       = 3  COLLAGEN DISEASES
C                       = 4  DERMATOLOGY
C                       = 5  CHEMICAL OF PHYSICAL AGENTS
C                       = 6  ENDOCRINOLOGY AND METABOLISM
C                       = 7  GASTROINTESTINAL (INCLUDING LIVER,PANCREAS)
C                       = 8  HEMATOLOGY
C                       = 9  INFECTIOUS DISEASES
C                       =10  MUSCULOSKELETAL
C                       =11  NEUROLOGY
C                       =12  PSYCHOLOGICAL MEDICINE
C                       =13  PULMONARY
C                       =14  RENAL
C                       =15  THERAPEUTICS
C                       =16  ANATOMY
C                       =17  BIOCHEMISTRY
C                       =18  GENETICS
C                       =19  LABORATORY MEDICINE
C                       =20  MICROBIOLOGY
C                       =21  PATHOLOGY
C                       =22  PHARMACOLOGY
C                       =23  PHYSIOLOGY
```

```
C        NTYP  - TYPE OF QUESTION
C              - IF NTYP=1  SINGLE ANSWER
C                       =2  MULTIPLE ANSWER
C        NTAX  - TAXONOMIC LEVEL
C              - IF NTAX=1  FACTUAL
C                       =2  PROBLEM SOLVING
C        NCORE - CORE LEVEL
C              - IF CORE=1  ESSENTIAL MATERIAL
C                       =2  MORE IMPORTANT THAN UNIMPORTANT MATERIAL
C                       =3  MORE UNIMPORTANT THAN IMPORTANT MATERIAL
C        NUM   - NUMBER OF ITEMS DESIRED WITH THESE RESTRICTIONS
C        ITR   - HIERARCHICAL
C              - IF  ITR=1  ALL LEVELS IF NECESSARY
                     ITR=2  ONLY THREE LEVELS
                     ITR=3  ONLY TWO LEVELS
                     ITR=4  FIRST LEVEL ONLY
C
C     REMARKS
C        LIMITATIONS
C              -MAX 100 CARDS PER ITEM
C              -MAX 9999 ITEMS
C
C     SUBROUTINES
C        SORT
C        PARMTR
C        TEXT
C
      DIMENSION STEM(100,17),NQ(100),NPT(100),IX(22),FX(33),NAM(2,23),NC
     1H(2,2),NTA(1,3),NES(1,3)
      DATA NAM/'ALL ','    ','CVS ','    ','COL ','    ','DERM','    ','
     1PHYS','CHEM','END ','MET ','GI  ','    ','HEMA','T   ','INF ','
     2 ','SKEL','    ','NEUR','    ','PSYC','    ','PULM','    ','REN ',
     3'    ','THER','    ','ANAT','    ','BIOC','    ','GEN ','    ','LA
     4BM','ED  ','MICR','OB  ','PATH','    ','PHAR','M   ','PHYS','IOL '
     5/,IO/'0'/,IBLK/' '/,IA/'*'/,IE/'E'/,NCH/'SING',' ANS','MULT',' ANS
     6'/,NTA/'FACT','COMP','PROB'/,NES/'ESS.','IMP.','UIMP'/
C DEFINITIONS:
C    NCI    :   COUNTER FOR NUMBER OF ITEMS PRINTED
C    NITEM1:   COUNTER FOR NUMBER OF ITEMS MEETING FIRST 4 RESTRICTIONS
C    NITEM2:   COUNTER FOR NUMBER OF ITEMS MEETING FIRST 3 RESTRICTIONS
C    NITEM3:   COUNTER FOR NUMBER OF ITEMS MEETING FIRST 2 RESTIRCTIONS
C    NITEM4:   COUNTER FOR NUMBER OF ITEMS MEETING FIRST 1 RESTRICTIONS
      CALL PDISC
  100 CONTINUE
      NCI= 0
      NITEM1=0
      NITEM2=0
      NITEM3=0
      NITEM4=0
      NDISC=0
```

```
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
      READ(5,1)MED,NTYP,NTAX,NCORE,NUM,ITR
    1 FORMAT(6I5)
      IF(MED.EQ.0) GO TO 27
      WRITE(6,2)(MED,(NAM(I,MED),I=1,2),NTYP,(NCH(I,NTYP),I=1,2),NTAX,(N
     1TA(I,NTAX),I=1,1),NCORE,(NES(I,NCORE),I=1,1),NUM)
    2 FORMAT(1H1,10X,'RESTRICTIONS IMPOSED:          '//,15X,'AREA OF SU
     1BSPECIALTY'10X,I2,2X,'- ',2A4,/,15X,'TYPE OF QUESTION',15X,I1,2X,'
     2- ',2A4,/,15X,'TAXONOMIC LEVEL',16X,I1,2X,'- ',1A4,/,15X,'CORE LEV
     3EL',21X,I1,2X,'- ',1A4,/,15X,'NUMBER OF ITEMS REQUESTED',3X,I4,//)
    3 CONTINUE
      J=0
    4 CONTINUE
C READ # OF CARDS CONTAINING ITEM(STEM(J,K))
      J=J+1
      READ(1)(STEM(J,K),K=1,17),NQ(J),NPT(J)
      IF(NPT(J).EQ.IE) GO TO 22
C END OF ITEM SENSED IF NPT(J) IS BLANK. THEN READS TWO PARAMETER CARDS
C ACCOMPANYING EACH ITEM.
      IF(NPT(J).NE.IBLK)  GO TO 4
      READ(1)(IX(I),I=20,22),(FX(K),K=1,16)
      READ(1)(FX(K),K=17,33)
C CHECK FOR MATACH OF ITEM ID#(NQ(J)) AND FIRST PARAMETER CARD ID # (ID1)
      IF(IX(1).EQ.MED) GO TO 7
      GO TO 3
    7 CONTINUE
C IF ITEM MEETS ONLY FIRST RESTRICTION WRITE IT ON TAPE 4
C AND INCREMENT NITEM4
      IF(IX(2).EQ.NTYP) GO TO 11
      IF(ITR.GT.1) GO TO 3
      NITEM4=NITEM4+1
      DO 9 L=1,J
      WRITE(4)(STEM(L,K),K=1,17),NQ(L),NPT(L)
    9 CONTINUE
      WRITE(4)(IX(I),I=1,19)
      WRITE(4)(IX(I),I=20,22),(FX(K),K=1,16)
      WRITE(4)(FX(K),K=17,33)
   10 CONTINUE
      GO TO 3
   11 CONTINUE
C IF ITEM MEETS ONLY FIRST TWO RESTRICTIONS WRITE ON TAPE 3
C AND INCREMENT NITEM3
      IF(IX(3).EQ.NTAX) GO TO 15
      IF(ITR.GT.2) GO TO 3
      NITEM3=NITEM3+1
      DO 13 L=1,J
      WRITE(3)(STEM(L,K),K=1,17),NQ(L),NPT(L)
```

```
   13 CONTINUE
      WRITE(3)(IX(I),I=1,19)
      WRITE(3)(IX(I),I=20.22),(FX(K),K=1,16)
      WRITE(3)(FX(K),K=17,33)
   14 CONTINUE
      GO TO 3
   15 CONTINUE
C IF ITEM MEETS ONLY FIRST THREE RESTRICTIONS WRITE ON TAPE 2
C AND INCREMENT NITEM2
      IF(IX(4).EQ.NCORE) GO TO 19
      IF(ITR.GT.3) GO TO 3
      NITEM2=NITEM2+1
      DO 17 L=1,J
      WRITE(2)(STEM(L,K),K=1,17),NQ(L),NPT(L)
   17 CONTINUE
      WRITE(2)(IX(I),I=1,19)
      WRITE(2)(IX(I),I=20,22),(FX(K),K=1,16)
      WRITE(2)(FX(K),K=17,33)
   18 CONTINUE
      GO TO 3
   19 CONTINUE
C IF ITEM MEETS ALL FOUR RESTRICTIONS PRINT AND INCREMENT NITEM1
      NITEM1=NITEM1+1
      IF(NITEM1.NE.1) GO TO 29
      WRITE(6,28)
   28 FORMAT(1H0,30X,'THE FOLLOWING ITEMS MEET THE ABOVE RESTRICTIONS:',
     1/////)
   29 CONTINUE
      NCI=NCI + 1
      WRITE(6,33) NCI,(STEM(1,K),K=1,17),NQ(1)
   33 FORMAT(2X,I4,'.',3X,17A4,10X,'ITEM ID IS:',I4)
C IF NPT(J) IS A * SKIP A LINE IN PRINTING
      IF(NPT(1).EQ.IC) GO TO 35
      WRITE(6,36)
   36 FORMAT(1H )
   35 CONTINUE
      DO 21 L=2,J
      WRITE(6,20)(STEM(L,K),K=1,17)
   20 FORMAT(10X,17A4)
      IF(NPT(L).EQ.IX) GO TO 21
      WRITE(6,101)
  101 FORMAT(1H )
   21 CONTINUE
      CALL PARMTR (IX,FX)
   38 CONTINUE
      CALL TEXT
      WRITE(6,32)
   32 FORMAT(1H1)
      GO TO 3
```

```
   22 CONTINUE
      IF(NITEM1.GE.NUM)GO TO 37
      IF(ITR.EQ.4) GO TO 37
      REWIND 2
      REWIND 3
      REWIND 4
C TO OBTAIN AS MANY ITEMS AS REQUESTED(NUM), CHECK FOR HOW MANY SCRATCH
C DISCS(NUMBERED 2 TO 4, EACH OF WHICH CONTAINS ONLY FIRST 3,2,OR 1
C RESTRICTION RESPECTIVELY. READ THESE DISCS IN SORT
      NTOT=NITEM1+NITEM2
      NTOTA=NITEM3+NTOT
      NTOTAL=NITEM4+NTOTA
      IF(NTOT.GE.NU.) GO TO 24
      IF(ITR.EQ.3) GO TO 24
      IF(NTOTA.GE.NUM)GO TO 25
      IF(ITEM.EQ.2) GO TO 25
      NDISC=4
      GO TO 26
   24 CONTINUE
      NDISC=2
      GO TO 26
   25 CONTINUE
      NDISC=3
   26 CONTINUE
      CALL SORT (NDISC,NITEM1,NITEM2,NITEM3,NITEM4,NUM,IC,IBLK,IA,IE,NTO
     1T,NTOTA,NTOTAL,NCI)
   37 CONTINUE
      GO TO 100
   27 CONTINUE
      REWIND 8
      STOP
      END
```

```
      SUBROUTINE PDISC
      DIMENSION STEM(100,17),NQ(100),NPT(100),IX(22),FX(33)
      DATE IC/'C'/,IBLK/' '/,IA/'*'/,IE/'E'/
      REWIND 8
    3 CONTINUE
      J=0
    4 CONTINUE
C READ # OF CARDS CONTAINING ITEM(STEM(J,K))
      J=J+1
      READ(8,5)(STEM(J,K),K=1,17),NQ(J),NPT(J)
    5 FORMAT(1X,17A4,4X,I4,2X,A1)
      IF(J-1)63,63,59
   59 CONTINUE
C CHECK FOR CONSISTENT ITEM ID # (NQ(J))
      IF(NQ(J)-NQ(J-1))60,62,60
   60 CONTINUE
      WRITE(6,61)NQ(J),NQ(J-1)
   61 FORMAT(/////,1X,'****NOTE,NOTE,NOTE,NOTE****',/,6X,'MISMATCH OF ID
     1S WITHIN ITEM:',I4,' AND ',I4,/////)
   62 CONTINUE
   63 CONTINUE
C CHECK FO? INVALID CHARACTER IN NPT(J). ALLOW C,E,*, OR BALNK
      IF(NPT(J).NE.IC.AND.NPT(J).NE.IBLK.AND.NPT(J).NE.IA.AND.NPT(J).NE.
     1IE)GO TO 41
      GO TO 43
   41 CONTINUE
      WRITE(6,42) NQ(J), NPT(J), J
   42 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE NOTE',/,6X,'ITEM ',I4,'
     1HAS THIS INCORRECT ALPHABETIC ENDING:(',A1,') ON CARD',I3,/////)
   43 CONTINEU
C IF E END OF TAPE IS SENSED
      IF(NPT(J).EQ.IE) GO TO 22
C END OF ITEM SENSED IF NPT(J) IS BLANK. THEN READS TWO PARAMETER CARDS
C ACCOMPANYING EACH ITEM.
      IF(NPT(J).NE.IBLK)  GO TO 4
      READ(8,6)(IX(I),I=1,22),(FX(K),K=1,16),NCARD1,ID1,(FX(K),K=17,33),
     1NCARD2,ID2
    6 FORMAT(I2,3I1,2(I2,I1),I3,5I1,2I1,2I2,2I1,I3,I4,16F2.2,9X,I1,I4,/,
     117F2.2,41X,I1,I4)
C CHECK FOR MATCH OF ITEM ID#(NQ(J)) AND FIRST PARAMETER CARD ID # (ID1)
      IF(NQ(J)-ID1)56,58,56
   56 CONTINUE
      WRITE(6,57) NQ(J),ID1
   57 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE****',/,6X,'MISMATCH OF ID
     1S BETWEEN ITEM AND PARMETER CARD:',/,6X,'ITEM ID IS:',I4,/,6X,'PAR
     1AMETER ID IS:',I4,/////)
      GO TO 3
   58 CONTINUE
C CHECK FOR ORDERING OF PARAMETER CARDS (NCARD1 AND NNARD2)
```

```
      IF(NCARD2-NCARD1)50,50,52
   50 CONTINUE
      WRITE(6,51) NCARD1,ID1,NCARD2,ID2
   51 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE ',/,5X,'PARAMETER CARDS A
     1RE OUT OF ORDER.',/,5X,'FIRST PARMETER CARD READS:CARD ',I1,2X'ITE
     2M # ',I4,/,5X,'SECOND PARAMETER CARD READS: CARD ',I1,2X'ITEM #',I
     34,/////)
      GO TO 3
   52 CONTINUE
C CHECK FOR MATCH OF PARAMETER CARDS ID #'S (ID1. AND ID2)
      IF(ID1-ID2)53,54,53
   53 CONTINUE
      WRITE(6,51) NCARD1,ID1,NCARD2,ID2
      GO TO 3
   54 CONTINUE
C IF ITEM DOES NOT MEET FIRST RESTRICTION READ NEXT ITEM
      DO 9 L=1,J
      WRITE(1)(STEM(L,K),K=1,17),NQ(L),NPT(L)
    9 CONTINUE
      WRITE(1)(IX(I),I=1,19)
      WRITE(1)(IX(I),I=20,22),(FX(K),K=1,16)
      WRITE(1)(FX(K),K=17,33)
      GO TO 3
   22 CONTINUE
      WRITE(1)(STEM(1,K),K=1,17),NQ(1),NPT(1)
      RETURN
      END
```

```
      SUBROUTINE SORT(NDISC,NITEM1,NITEM2,NITEM3,NITEM4,NUM,IC,IBLK,IA,I
     1E,NTOT,NTOTA,NTOTAL,NCI)
      DIMENSION IX(22),STEM(100,17),NQ(100),NPT(100),ND(100),FX(33)
C SIZE OF DO LOOP DETERMINED BY # OF ITEMS NEEDED
      DO 38   I=2,NDISC
      NCRT=0
C START OF EACH NEW DISC THESE STATEMENTS ARE WRITTEN
      IF (I.E.Q.2) GO TO  6
      IF (I.E.Q.3) GO TO 11
      IF (I.E.Q.4) GO TO 16
    6 CONTINUE
      WRITE(6,2) NITEM1
    2 FORMAT(1H0,29X'THE ABOVE ',I4,"   ITEMS MET THE FOUR RESTRICTIONS
     1SUBMITTED.',/,30X,'THE NUMBER OF THESE ITEMS IS LESS THAN THE QUOT
     2A ASKED.')
      WRITE(6,7)
    7 FORMAT(1H ,29X'TO MEET QUOTA,ITEMS MEETING THE FIRST THREE RESTRIC
     1TIONS ARE GIVEN.')
      IF(NTOT.G.E.NUM)GO TO 9
      WRITE(6,8)
    8 FORMAT(30X,'HOWEVER,THIS IS STILL LESS THAN THE QUOTA.'/////)
      GO TO 21
    9 CONTINUE
      WRITE(6,10)
   10 FORMAT(30X'WITH THIS SECOND SET OF ITEMS THE QUOTA IS MET OR EXCEE
     1DED.',/////)
      GO TO 21
   11 CONTINUE
      WRITE(6,12)
   12 FORMAT(1H1,29X,'TO MEET QUOTA,ITEMS MEETING THE FIRST TWO RESTRICT
     1IONS ARE NOW ALSO GIVEN.'   )
      IF(NTOTA.GE.NUM)GO TO 14
      WRITE(6,13)
   13 FORMAT(30X,'HOWEVER,THIS IS AGAIN LESS THAN THE QUOTA.' /////)
      GO TO 21
   14 CONTINUE
      WRITE(6,15)
   15 FORMAT(30X,'WITH THIS THIRD SET OF ITEMS THE QUOTA IS MET OR EXCEE
     1DED.'/////)
      GO TO 21
   16 CONTINUE
      WRITE(6,17)
   17 FORMAT(1H1,29X,'TO MEET QUOTA,ITEMS MEETING THE FIRST RESTRICTION
     1ARE NOW ALSO GIVEN.'   )
      IF(NTOTAL.GE.NUM)GO TO 19
      WRITE(6,18)
   18 FORMAT(30X'HOWEVER, THE QUOTA CANNOT BE MET WITH THE PRESENT STORAG
     1E OF ITEMS.' /////)
      GO TO 21
   19 CONTINUE
      WRITE (6,20)
   20 FORMAT(30X,'WITH THIS FOURTH SET OF ITEMS THE QUOTA IS MET OR EXCE
     1EDED.' /////)
   21 CONTINUE
```

```
C CHECK TO SEE IF WRITTEN OUT ALL ITEMS PUT ON RESPECTIVE
C DISC.
      IF(I.EQ.2)GO TO 24
      IF(I.EQ.3)GO TO 25
      IF(I.EQ.4)GO TO 26
   24 CONTINUE
      IF(NCRT.EQ.NITEM2)GO TO 37
      GO TO 27
   25 CONTINUE
      IF(NCRT.EQ.NITEM3)GO TO 37
      GO TO 27
   26 CONTINUE
      IF(NCRT.EQ.NITEM4)GO TO 37
      GO TO 27
   27 CONTINUE
      J=0
   28 CONTINUE
C READ AND WRITE ITEM IN SAME MANNER AS IN MAIN LINE.
      J=J+1
      READ(I)(STEM(J,K),K=1,17),NQ(J),NPT(J)
      IF(NPT(J).NE.IBLK) GO TO 28
      READ(I) (IX(NI),NI=1,19)
      READ(I) (IX(NI),NI-20,22),(FX(KK),KK=1,16)
      READ(I) (FX(KK),KK=17,33)
      NCI=NCI + 1
      WRITE(6,40) NCI,(STEM(1,K),K=1,17),NQ(1)
   40 FORMAT(2X,I4,'.',3X,17A4,10X,'ITEM ID IS:',I4)
      IF(NPT(1).EQ.IA) GO TO 41
      GO TO 43
   41 CONTINUE
      WRITE(6,42)
   42 FORMAT(1H )
   43 CONTINUE
      DO 34 L=],J
      WRITE(6,31)(STEM(L,K),K=1,17)
   31 FORMAT(10X,17A4)
      IF(NPT(L).E.Q.IA) GO TO 32
      GO TO 34
   32 CONTINUE
      WRITE(6,33)
   33 FORMAT(1H )
   34 CONTINUE
      CALL PARMTR (IX,FX)
   36 CONTINUE
      CALL TEXT
      NCRT=NCRT+1
      WRITE(6,22)
   22 FORMAT(1H1)
      GO TO 21
   37 CONTINUE
   38 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PARMTR(IX,FX)
      DIMENSION NAM(2,23),NCH(2,2),NTA(1,3),NES(1,3),NAREA(1,5),NPROV(1,
     116),NVIDO(2,6),NGRAD(2,2),NAT(2,2),NLANG(2,3),IX(22),FX(33)
      DATA NAM/'ALL ','    ','CVS ','    ','COL ','    ','DERM','    ','
     1PHYS','CHEM','END ','MET ','GI  ','    ','HEMA','T   ','INF ','
     2 ','SKEL','    ','NEUR','    ','PSYC','    ','PULM','    ','REN ',
     3'    ','THER','    ','ANAT','    ','BIOC','    ','GEN ','    ','LA
     4BM','ED  ','MICR','OB  ','PATH'm'    ','PHAR','M   ','PHYS','IOL '
     5/
      DATA NCH/'SING',' ANS','MULT'n' ANS'/
      DATA NTA/'FACT','COMP','PROB'/
      DATA NES/'ESS.','IMP.','UIMP'/
      DATA NAREA/'AMIB','NBME','CAN ','UK  ','OTH '/
      DATA NPROV/'ALTA','B.C.','DALH','LAVL','MCG ','MCM ','MAN ','MTRL'
     1,'OTT ','QN  ','SASK','SHRB','TOR ','UWO ','CALG','MMRL'/
      DATA NVIDO/'LINE','    ','PHOT','O   ','COLO','R   ','SLID','E   '
     1,'MOVI','E   ','VIDE','O   '/
      DATA NGRAD/'GRAD','.   ','UGRA','D.  '/
      DATA NAT/'NAT.','EXAM','LCC.','EXAM'/
      DATA NLANG/'BTH.','LANG'.'ENG.'.'ONLY'.'FR. ','ONLY'/
C THIS SUBROUTINE CHECKS THE PARAMETER CARDS ACCOMPANYING
C EACH ITEM AND SUBSTITUTES THE NUMERIC VALUES WITH PROPER
C ALPHABETIC NAME.(SOME NUMERALS ARE PRINTED.)
      WRITE(6,37)
   37 FORMAT(///,39X,'PARAMETERS FOR THIS ITEM:',37X,'REVIEWER PLEASE CO
     1MMENT ON:',//)
      J=IX(1)
      WRITE(6,1)(NAME(I,J),I=1,2)
    1 FORMAT(10X,'AREA OF SUBSPECIALTY:',2X,2A4,44X,'CONTENT',20X,'OK?..
     1..')
      J=IX(5)
      IF(J.LE.0.OR.J.GT.23) GO TO 3
      WRITE(6,2) (NAME(I,J),I=1,2)
    2 FORMAT(10X,'SECOND AREA OF SUBSPECIALTY:',2X,2A4,44X,'SENSE',22X,'
     1OK?....')
      GO TO 41
    3 CONTINUE
      WRITE(6,42)
   42 FORMAT(10X,'SECOND AREA OF SUBSPECIALTY:',54X,'SENSE',22X,'OK?....
     1')
   41 CONTINUE
      J=IX(2)
      IF(J.LE.0.OR.J.GT.3)  GO TO 5
      WRITE(6,4) (NCH(I,J),I=1,2)
    4 FORMAT(10X,'TYPE OF QUESTION:',13X,2A4,44X,'GRAMMAR",20X,'OK?....'
     1)
    5 CONTINUE
      J=IX(3)
      IF(J.LE.0.OR.J.GE.3)GO TO 7
      WRITE(6,6)(NTA(I,J),I=1,1)
```

```
 6 FORMAT(10X,'TAXONOMIC LEVEL:',14X,1A4,48X,'RESPONSES CORRECT',10X,
  1'OK?....')
 7 CONTINUE
   J=IX(4)
   IF(J.LE.0.OR.J.GT.3) GO TO 9
   WRITE(6,8) (NES(I,J),I=1,1)
 8 FORMAT(10X,'CORE LEVEL:',19X,1A4,48X,'DISTRACTORS',16X,'OK?....')
   GO TO 43
 9 CONTINUE
   WRITE (6,44)
44 FORMAT(10X,'CORE LEVEL:',71X,'DISTRACTORS',16X,'OK?....')
43 CONTINUE
   J=IX(6)
   IF(J.LE.0.OR.J.GT.5) GO TO 11
   WRITE(6,10) (NAREA(I,J),I=1,1)
10 FORMAT(10X,'SOURCE:',23X,1A4,48X,'WRITTEN COMMENTS:')
   GO TO 46
11 CONTINUE
   WRITE(6,45)
45 FORMAT(10X,'SOURCE:',75X,'WRITTEN COMMENTS:')
46 CONTINUE
   J=IX(7)
   IF(J.LE.0.OR.J.GT.16) GO TO 13
   WRITE(6,12) (NPOV(I,J),I=1,1)
12 FORMAT(10X'UNIVERSITY:',19X,1A4)
13 CONTINUE
   J=IX(8)
   IF(J.LE.0.OR.J.GT.6) GO TO 15
   WRITE(6,14)(NVIDO(I,J),I=1,2),IX(9)
14 FORMAT(10X'AUDIO VISUAL MATERIAL NEEDED: ',2A4,': IDENT.OF SOURCE:
  1 ',I3,/)
15 CONTINUE
   WRITE(6,16)
16 FORMAT(10X'CORRECT RESPONSE ALTERNATIVE:')
   WRITE(6,17)
17 FORMAT(10X'CHOICE 1  CHOICE 2  CHOICE 3  CHOICE 4  CHOICE 5 ' )
   WRITE(6,18)(IX(I),I=10,14)
18 FORMAT(13XI1, 9X,I1, 9X,I1, 9X,I1, 9X,I1,/)
   J=IX(15)
   IF(J.LE.0.OR.J.GT.3)GO TO 31
   WRITE(6,30)(NLANG(I,J),I=1,2)
30 FORMAT(10X'THE QUESTION IS AVAILABLE IN ',2A4,/)
31 CONTINUE
   WRITE(6,23) IX(16)
23 FORMAT(10X'NUMBER OF TIMES USED:',10X,I1)
   IF(IX(16).EQ.0)  GO TO 40
   IF(IX(17).EQ.0)  GO TO 20
   WRITE(6,19) IX(17)
19 FORMAT(10X'LAST YEAR QUESTION WAS USED:',2X,I2)
20 CONTINUE
   IF(IX(18).EQ.0) GO TO 58
   WRITE(6,22)IX(18)
```

```
22 FORMAT(10X,'QUESTION NUMBER ON LAST EXAM:',1X,I2,/)
58 CONTINUE
   J-IX(19)
   IF(J.LE.0.OR.J.GT.2) GO TO 27
   WRITE(6,26) (NAT(I,J),I=1,2)
26 FORMAT(53X,2A4)
27 CONTINUE
   IF(IX(21).EQ.0) GO TO 29
   WRITE(6,28) IX(21)
28 FORMAT(53X,'ID=',I3)
29 CONTINUE
   IF(IX(22).EQ.0) GO TO 59
   WRITE(6,47)(FX(K),K=1,2)
47 FORMAT(10X,'DIFFICULTY LEVEL OF THIS SINGLE-CORRECT-ANSWER TYPE OF
  1 QUESTION',/,10X,'AT THESE RECORDED TESTING YEARS:',/,10X,'FIRST
  2  SECOND',/,11X,F4.2,6X,F4.2,/)
   WRITE(6,48)(FX(K),K=3,4)
48 FORMAT(10X,'BISERIAL COEFFICIENT FOR THESE RECORDED TESTING YEARS:
  1',/,10X'FIRST    SECOND',/,11X,F4.2,6X,F4.2,/)
   GO TO 54
49 CONTINUE
   WRITE(6,50)
50 FORMAT(10X,'DIFFICULTY LEVELS AND BISERIAL COEFFICIENTS OF THIS MU
  1LTIPLE-ANSWER TYPE OF',/,10X,'QUESTION FOR THE FIRST RECORDED TEST
  2ING YEAR:')
   WRITE(6,51)
51 FORMAT(15X'1ST CHOICE  2ND CHOICE  3RD CHOICE  4TH CHOICE  5TH CHO
  1ICE  TOTAL ITEM')
   WRITE(6,52)(FX(K),K=5,16)
52 FORMAT(10X,3H'P',6X,6(F4.2,8X),/,10X,3H'R',6X,6(F4.2,8X),/)
   IF(IX(16).LT.2) GO TO 54
   WRITE(6,53)
53 FORMAT(10X,'DIFFICULTY LEVELS AND BISERIAL COEFFICIENTS OF THIS MU
  1LTIPLE-ANSWER TYPE OF ',/,10X,'QUESTION FOR THE SECOND RECORDED TE
  2STING YEAR:')
   WRITE(6,51)
   WRITE(6,52)(FX(K),K-17,28)
54 CONTINUE
   WRITE(6,55)(FX(K),K=29,33)
55 FORMAT(10X,'PROPORTION ON LAST TEST SELECTING THIS DISTRACTOR:',/,
  110X,'FIRST  SECOND  THIRD  FOURTH  FIFTH',/,11X,F4.2,4X,F4.2,3X,F4
  2.2,4X,F4.2,3X,F4.2)
40 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE TEXT
C THIS SUBROUTINE SIMPLY WRITES OUT A CHECK LIST FOR
C TEST COMMITTEE
      WRITE(6,1)
    1 FORMAT(//)
      WRITE(6,3)
    3 FORMAT(10X,'TEST COMMITTEE:',/,26X,'ACCEPT AS IS:',24X,'YES.... NO
     1....',/,26X,'MODIFY AS NOTED AND ACCEPT:',10X,'YES.... NO....',/,2
     26X,'CATEGORY A....  B....',16X,'YES.... NO....',/,26X,'DATE.......
     3...',/,26X,'INITIAL.......')
      WRITE(6,30)
   30 FORMAT(1H0,25X,'...............................................
     1.')
      RETURN
      END
```

A P P E N D I X C

Program UTILITY: Stack Tape

UTILITY PROGRAM: CARD TO TAPE

To <u>STACK</u> tape: prepare these system cards and place before data:

```
col.1    col.12
  ↓        ↓
1) //PRINT    EXEC PGM=IEBGENER
2) //SYSPRINT DD SYSOUT=A
         col.12
           ↓
3) //SYSIN    DD DUMMY                                   col.72
                                                            ↓
4) //SYSUT2 DD DSNAME=MEDS,LABEL=(1,SL),UNIT=SYSTP,         C
             col.16
               ↓
5) //          DCB=(RECFM=FB,BLKSIZE=7200,LRECL=80,DEN=2),  C
                          VB
                          F
                          V
                          U
             col.16
               ↓
6) //          VOLUM=SER=T0025,DISP=(NEW,KEEP)
7) //SYSUT1 DD *
8) data cards to be stacked
9) /*
```

A P P E N D I X   D

Program CHECK

```
      DIMENSION STEM(100,17),NQ(100),NPT(100),IX(22),FX(33)
      DATE IC/'C'/,IBLK/' '/,IA/'*'/,IE/'E'/
    3 CONTINUE
      J=0
    4 CONTINUE
C READ # OF CARDS CONTAINING ITEM(STEM(J,K))
      J=J+1
      READ(5,5)(STEM(J,K),K=1,17),NQ(J),NPT(J)
    5 FORMAT(1X,17A4,4X,I4,2X,A1)
      IF(J-1)63,63,59
   59 CONTINUE
C CHECK FOR CONSISTENT ITEM ID # (NQ(J))
      IF(NQ(J)-NQ(J-1))60,62,60
   60 CONTINUE
      WRITE(6,61)NQ(J),NQ(J-1)
   61 FORMAT(/////,1X,'****NOTE,NOTE,NOTE,NOTE****',/,6X,'MISMATCH OF ID
     1S WITHIN ITEM:',I4,' AND ',I4,/////)
   62 CONTINUE
   63 CONTINUE
C CHECK FOR INVALID CHARACTER IN NPT(J). ALLOW C,E,*, OR BALNK
      IF(NPT(J).NE.IC.AND.NPT(J).NE.IBLK.AND.NPT(J).NE.IA.AND.NPT(J).NE.
     1IE)GO TO 41
      GO TO 43
   41 CONTINUE
      WRITE(6,42) NQ(J), NPT(J), J
   42 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE NOTE',/,6X,'ITEM ',I4,'
     1HAS THIS INCORRECT ALPHABETIC ENDING:(',A1,') ON CARD',I3,/////)
   43 CONTINEU
C IF E END OF TAPE IS SENSED
      IF(NPT(J).EQ.IE) GO TO 22
C END OF ITEM SENSED IF NPT(J) IS BLANK. THEN READS TWO PARAMETER CARDS
C ACCOMPANYING EACH ITEM.
      IF(NPT(J).NE.IBLK)  GO TO 4
      READ(5,6)(IX(I),I=1,22),(FX(K),K=1,16),NCARD1,ID1,(FX(K),K=17,33),
     1NCARD2,ID2
    6 FORMAT(I2,3I1,2(I2,I1),I3,5I1,2I1,2I2,2I1,I3,I4,16F2.2,9X,I1,I4,/,
     117F2.2,41X,I1,I4)
C CHECK FOR MATCH OF ITEM ID#(NQ(J)) AND FIRST PARAMETER CARD ID # (ID1)
      IF(NQ(J)-ID1)56,58,56
   56 CONTINUE
      WRITE(6,57) NQ(J),ID1
   57 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE****',/,6X,'MISMATCH OF ID
     1S BETWEEN ITEM AND PARMETER CARD:',/,6X,'ITEM ID IS:',I4,/,6X,'PAR
     1AMETER ID IS:',I4,/////)
      GO TO 3
   58 CONTINUE
C CHECK FOR ORDERING OF PARAMETER CARDS (NCARD1 AND NNARD2)
```

```
      IF(NCARD2-NCARD1)50,50,52
   50 CONTINUE
      WRITE(6,51) NCARD1,ID1,NCARD2,ID2
   51 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE ',/,5X,'PARAMETER CARDS A
     1RE OUT OF ORDER.',/,5X,'FIRST PARMETER CARD READS:CARD ',I1,2X'ITE
     2M # ',I4,/,5X,'SECOND PARAMETER CARD READS: CARD ',I1,2X'ITEM #',I
     34,/////)
      GO TO 3
   52 CONTINUE
C CHECK FOR MATCH OF PARAMETER CARDS ID #'S (ID1. AND ID2)
      IF(ID1-ID2)53,54,53
   53 CONTINUE
      WRITE(6,51) NCARD1,ID1,NCARD2,ID2
      GO TO 3
   22 CONTINUE
      RETURN
      END

***** END OF COMPILATION *****
```

A P P E N D I X   E

Program MEDSIRCH-2

```
C
C     MEDSIRCH2    DIVISION OF EDUCATIONAL RESEARCH SERVICES
C                              UNIVERSITY OF ALBERTA
C     ..................................................................
C
C                  SEARCH FOR MEDICAL EXAMINATION QUESTIONS
C                  ROYAL COLLEGE OF PHYSICIANS AND SURGEONS
C                       DEPARTMENT OF INTERNAL MEDICINE
C
C     PROGRAMMER:  C.B.HAZLETT
C
C     PURPOSE:
C        1. READS MULTIPLE CHOICE ITEMS FROM TAPE SELECTING
C           THOSE MEETING THE FIRST FOUR RESTRICTIONS USER REQUIRES.
C        2. SELECTION DETERMINED BY ID NUMBER OF ITEM ACCOMPANYING
C           EACH PARAMETER CARD.
C
C     CARD INPUT:
C        1. PARAMETER CARD
C        2. ID NUMBERS (16I5)
C           SUBMIT CARDS 1 AND 2 IN PAIRS,
C           OR IF MORE THAN ONE CARD IS NEEDED
C           FOR ID NUMBERS, SUBMIT IN GROUPS.
C        3. LAST CARD: BLANK CARD
C
C     DESCRIPTION OF PARAMETER CARD (6I5)
C        LID   - NUMBER OF THIS CARD (MATCHES LED)
C        MED   - AREA OF SUBSPECIALITY
C              - IF MED= 1  ALLERGY,IMMUNOLOGY,SEROLOGY
C                      = 2  CARDIOVASCULAR
C                      = 3  COLLAGEN DISEASES
C                      = 4  DERMATOLOGY
C                      = 5  CHEMICAL OF PHYSICAL AGENTS
C                      = 6  ENDOCRINOLOGY AND METABOLISM
C                      = 7  GASTROINTESTINAL (INCLUDING LIVER,PANCREAS)
C                      = 8  HEMATOLOGY
C                      = 9  INFECTIOUS DISEASES
C                      =10  MUSCULOSKELETAL
C                      =11  NEUROLOGY
C                      =12  PSYCHOLOGICAL MEDICINE
C                      =13  PULMONARY
C                      =14  RENAL
C                      =15  THERAPEUTICS
C                      =16  ANATOMY
C                      =17  BIOCHEMISTRY
C                      =18  GENETICS
C                      =19  LABORATORY MEDICINE
C                      =20  MICROBIOLOGY
C                      =21  PATHOLOGY
C                      =22  PHARMACOLOGY
C                      =23  PHYSIOLOGY
```

```
C        NTYP  - TYPE OF QUESTION
C              - IF NTYP=1  SINGLE ANSWER
C                       =2  MULTIPLE ANSWER
C        NTAX  - TAXONOMIC LEVEL
C              - IF NTAX=1  FACTUAL
C                       =2  PROBLEM SOLVING
C        NCORE - CORE LEVEL
C              - IF CORE=1  ESSENTIAL MATERIAL
C                       =2  MORE IMPORTANT THAN UNIMPORTANT MATERIAL
C                       =3  MORE UNIMPORTANT THAN IMPORTANT MATERIAL
C        NUM   - NUMBER OF ITEMS DESIRED WITH THESE RESTRICTIONS
C
C     DESCRIPTION OF ID NUMBERS CARD(S)  (16I5)
C        LED   - NUMBER OF THIS SET (MATCHES LID)
C        NUMID - ID NUMBERS OF ITEMS TO BE SELECTED
C              - PUNCH IN FIELDS OF FIVE
C              - IF MORE THAN 15 ITEMS DESIRED FOR THIS SET
C                START ON NEXT CARD IN COLUMN 5
C
C     REMARKS
C        LIMITATIONS
C              -MAX 100 CARDS PER ITEM
C              -MAX 9999 ITEMS
C
C     SUBROUTINES
C        PDISC
C        PARMTR
C        TEXT
C
      DIMENSION STEM(100,17),NQ(100),NPT(100),IX(22),FX(33),NAM(2,23),NC
     1H(2,2),NTA(1,3),NES(1,3),NUMID(25)
      DATA NAM/'ALL ','    ','CVS ','    ','COL ','    ','DERM','    ','
     1PHYS','CHEM','END ','MET ','GI  ','    ','HEMA','T   ','INF ','
     2 ','SKEL','    ','NEUR','    ','PSYC','    ','PULM','    ','REN ',
     3'    ','THER','    ','ANAT','    ','BIOC','    ','GEN ','    ','LA
     4BM','ED  ','MICR','OB  ','PATH','    ','PHAR','M   ','PHYS','IOL '
     5/,IC/'C'/,IBLK/' '/,        IE/'E'/,NCH/'SING',' ANS','MULT',' ANS
     6'/,NTA/'FACT','COMP','PROB'/,NES/'ESS.','IMP.','UIMP'/
C DEFINITIONS:
C     NCI   "   COUNTER FOR NUMBER OF ITEMS PRINTED
      CALL PDISC
  100 CONTINUE
      NC1= 0
      REWIND 1
  108 CONTINUE
      READ(5,102)LID,MED,NTYP,NTAX,NCORE,NUM
  102 FORMAT(6I5)
      IF(LID.EQ.0)GO TO 27
      READ(5,103)LED,(NUMID(JI),JI=1,NUM)
  103 FORMAT(16I5)
      IF(LED.EQ.LID)GO TO 106
      WRITE(6,107)LID,LED
```

```
  107 FORMAT(1H0,15X,'MISMATCHED ID-PAIRS ON THESE PARAMETER CARDS : ',
     1I5,' AND ',I5)
      GO TO 108
  106 CONTINUE
      WRITE(6,2)(MED,(NAM(I,MED),I=1,2),NTYP,(NCH(I,NTYP),I=1,2),NTAX,(N
     1TA(I,NTAX),I=1,1),NCORE,(NES(I,NCORE),I=1,1),NUM)
     2FORMAT(1H1,10X,'RESTRICTIONS IMPOSED:          '//,15X,'AREA OF SU
     1BSPECIALTY'10X,I2,2X,'- ',2A4,/,15X,'TYPE OF QUESTION',15X,I1,2X,'
     2- ',2A4,/,15X,'TAXONOMIC LEVEL',16X,I1,2X,'- ',1A4,/,15X,'CORE LEV
     3EL',21X,I1,2X,'- ',1A4,/,15X,'NUMBER OF ITEMS REQUESTED',3X,I4,//)
    3 CONTINUE
      J=0
    4 CONTINUE
C READ # OF CARDS CONTAINING ITEM(STEM(J,K))
      J=J+1
      READ(1)(STEM(J,K),K=1,17),NQ(J),NPT(J)
      IF(NPT(J).EQ.IE) GO TO 100
C END OF ITEM SENSED IF NPT(J) IS BLANK.  THEN READS TWO PARAMETER CARDS
C ACCOMPANYING EACH ITEM.
      IF(NPT(J).NE.IBLK)  GO TO 4
      READ(1)(IX(I),I=1,19)
      READ(1)(IX(I),I=20,22),(FX(K),K=1,16)
      READ(10(FX(K),K=17,33)
      DO 104 JI=1,NUM
      IF(NQ(1).EQ.NUMID(JI)) GO TO 105
  104 CONTINUE
      GO TO 3
  105 CONTINUE
      NCI=NCI + 1
      IF(NCI.NE.1)GO TO 29
      WRITE(6,28)
   28 FORMAT(1H0,30X,'THE FOLLOWING ITEMS MEET THE ABOVE RESTRICTIONS:',
     1/////)
   29 CONTINUE
      WRITE(6,33) NCI,(STEM(1,K),K=1,17),NQ(1)
   33 FORMAT(2X,I4,'.',3X,17A4,10X,'ITEM ID IS:',I4)
C IF NPT(J) IS A * SKIP A LINE IN PRINTING
      IF(NPT(1).EQ.IC) GO TO 35
      WRITE(6,36)
   36 FORMAT(1H )
   35 CONTINUE
      DO 21 L=2,J
      WRITE(6,20)(STEM(L,K),K=1,17)
   20 FORMAT(10X,17A4)
      IF(NPT(L).EQ.IC) GO TO 21
      WRITE(6,101)
```

```
 101 FORMAT(1H )
  21 CONTINUE
     CALL TEXT
     WRITE(6,32)
  32 FORMAT(1H1)
     IF(NCI.GE.NUM)GO TO 100
 110 CONTINUE
     GO TO 3
  27 CONTINUE
     STOP
     END

***** END OF COMPILATION*****
```

SUBROUTINE PDISC
(see MEDSIRCH-1, Appendix B, p. 112

SUBROUTINE PARMTR
(see MEDSIRCH-1, Appendix B, p. 116)

SUBROUTINE TEXT
(see MEDSIRCH-1, Appendix B, p. 119)

A P P E N D I X   F

Program UPDATE

```
C
C     UPDATE          DIVISION OF EUDCATIONAL RESEARCH SERVICES
C                               UNIVERSITY OF ALBERTA
C     ...    .............................................................
C
C                      UPDATES TAPE BANK THAT WILL BE USED IN
C                     SEARCH FOR MEDICAL EXAMINATION QUESTIONS
C                     ROYAL COLLEGE OF PHYSICIANS AND SURGEONS
C                          DEPARTMENT OF INTERNAL MEDICINE
C
C     PROGRAMMER:  C.B.HAZLETT
C
C     PURPOSE:
C        1. DELETES UNWANTED ITEMS IN BANK.
C        2. REVISES INCORRECT PARAMETER CARDS FOR ITEMS.
C        3. ADDS NEW ITEMS.
C        4. DELETES THOSE ITEMS THAT ARE TO BE MODIFIED AND
C           ADDS CORRECTED ITEM.
C
C     CARD INPUT:
         1. TITLE CARD.      (20A4)
C        2. PARAMETER CARD   (3I5)
C        3. IDS OF ITEMS TO BE DELETED.(OPTIONAL)     (16I5)
C        4. PAIRS OF PARAMETER CARDS FOR ITEMS NEEDING
C           PARAMETER MODIFICATION. (OPTIONAL).
C        5. MULTIPLE CHOICE ITEMS AND ACCOMPANYING
C           PARAMETER CARDS THAT ARE TO BE ADDED.
C           INCLUDES ALL ITEMS BEING MODIFIED.
C
C     DESCRIPTION OF PARAMETER CARD (3I5)
C        ITEMRE   -NO. OF ITEMS TO BE DELETED.
C                 -INCLUDES THOSE NO LONGER WANTED
C                  AND THOSE BEING MODIFIED.
C        ITPARM   -NO. OF PAIRS OF PARAMETER CARDS
C                  BEING MODIFIED.
C        ITEMAD   -NO. OF ITEMS BEING ADDED.
C                 -INCLUDES THOSE NOW AND MODIFIED.
C
C     REMARKS
C        LIMITATIONS
C                 -MAX 1000 ITEMS DELETED.
C                 -MAX  600 PAIRS OF PARAMETER CARDS MODIFIED.
C                 -NO LIMIT ON NO. OF ITEMS ADDED
```

```
C
      DIMENSION STEM(100,17),NQ(100),NPT(100),IX(22),FX(33),IDRE(1000),
     1IXX(100,22),FXX(100,33),NCRD1( 100),NCRD2( 100),IDD1( 100),IDD2( 1
     200),TITLE(20)
      DATA IC/'C'/,IBLK/' '/,IA/'*'/,IE/'E'/
C     OLD TAPE IS 8, NEW TAPE IS 1
      REWIND 8
      REWIND 1
C     READ AND WRITE OUT TITLE CARD
      READ(5,200)TITLE
  200 FORMAT(20A4)
      WRITE(6,201) TITLE
  201 FORMAT(20X,20A4,///)
C     READ PARAMETER CARD
      READ(5,1)ITEMRE,ITPARM,ITEMAD
    1 FORMAT(3I5)
C     IF NOT REMOVING ANY ITEMS SKIP READING IN THE ID NO. FOR SUCH ITEMS
      IF(ITEMRE.EQ.0)GO TO 333
      READ(5,2)(IDRE(I),I=1,ITEMRE)
    2 FORMAT(16I5)
      WRITE(6,702)(IDRE(I),I=1,ITEMRE)
  702 FORMAT(09X,' THESE ITEMS HAVE BEEN REMOVED : ',I5,/,(42X,I5))
  333 CONTINUE
C     IF NO CHANGES IN PARAMETER SKIP READING CHANGES
      IF(ITPARM.EQ.0) GO TO 7
C     READ IN ALL PARAMETER CARDS THAT ARE USED AS MODIFICATIONS
      DO 102 N=1,ITPARM
      READ(5,103)(IXX(N,J),J=1,22),(FXX(N,K),K=1,16),NCRD1(N),IDD1(N),(F
     1XX(N,K),K=17,33),NCRD2(N),IDD2(N)
  103 FORMAT(I2,3I1,2(I2,I1),I3,5I1,2I1,2I2,2I1,I3,I4,16A2,oX,I1,I4,/,'7
     1A2,41X,I1,I4)
C     CHECK TO SEE THAT PAIR OF PARAMETER CARDS ARE IN CORRECT ORDER.
      IF(NCRD2(N)-NCRD1(N))80,80,82
   80 CONTINUE
      WRITE(6,81)NCRD1(N),IDD1(N),NCRD2(N),IDD2
   81 FORMAT(///,1X,,'*** NOTE:',/,5X,'THESE PARAMETER CARDS WHICH ARE B
     1EING USED AS MODIFICATIONS ARE OUT OF ORDER.',/,5X,'FIRST PARAMETE
     2R CARD READS:CARD ',I1,' ITEM NO. ',I4,/,5X,'SECOND PARAMETER CARD
     3 READS: CARD ',I1,' ITEM NO. ',I4,///)
C     COUNTER FOR MISTAKES MADE IN THIS RUN.
      NWRONG=NWRONG+1
   82 CONTINUE
      IF(IDD1(N)-IDD2(N))83,84,83
   83 CONTINUE
      WRITE(6,81)NCRD1(N),IDD1(N),NCRD2(N),IDD2(N)
      NWRONG=NWRONG+1
   84 CONTINUE
  102 CONTINUE
      WRITE(6,703)(IDD1(N),N=1,ITPARM)
  703 FORMAT(/,10X,'PARAMETER CARDS OF THESE ITEMS ( INDICATED BY THEIR
     1ID NUMBERS ) HAVE BEEN CHANGED:',I5,/,(93X,I5))
```

```
    7 CONTINUE
  105 CONTINUE
C     READ ITEMS AND PARAMETER CARDS FROM OLD TAPE
      J=0
  106 CONTINUE
      J=J+1
      READ(8,110)(STEM(J,K),K=1,17),NQ(J),NPT(J)
  110 FORMAT(1X,17A4,4X,I4,2X,A1)
C     IF COLUMN 80 HAS AN E ON OLD TAPE THE END IS SENSED.
      IF(NPT(J).EQ.IE)GO TO 122
C     IF COLUMN 80 HAS A BLANK THEN END OF THIS ITEM
      IF(NPT(J).NE.IBLK)GO TO 106
C     IF END OF ITEM READ PAIR OF PARAMETER CARDS
      READ(8,103)(IX(1),I=1,22),FX(K),K=1,16),NCARD1,ID1,(FX(K),K=17,33
     1),NCARD2,ID2
C     CHECK TO SEE IF ITEM READ FROM OLD TAPE IS TO BE REMOVED
      IF(ITEMRE.EQ.0)GO TO 120
      DO 107 M=1,ITEMRE
      IF(NQ(1)-IDRE(M))107,105,107
  107 CONTINUE
  120 CONTINUE
C     IF ITEM NOT REMOVED OR MODIFIED WRITE ON NEW TAPE
      DO 108 L=1,J
      WRITE(1,110)(STEM(L,K),K=1,17),NQ(L),NPT(L)
  108 CONTINUE
C     CHECK FOR CORRECTION OF PARAMETER CARD
      IF(ITPARM.EQ.0)GO TO 121
      DO 109 N=1,ITPARM
      IF(IDD1(N)-ID1) 109,111,109
  109 CONTINUE
      GO TO 121
  111 CONTINUE
C     WRITE CORRECTED PARAMETER CARDS.
      WRITE(1,103)(IXX(N,J),J=1,22),(FXX(N,K),K=1,16),NCRD1(N),IDD1(N),(
     1FXX(N,K),K=17,33),NCRD2(N),IDD2(N)
      GO TO 105
  121 CONTINUE
C     WRITE OLD PARAMETER CARDS
      WRITE(1,103)(IX(I),I=1,22),(FX(K),K=1,16),NCARD1,ID1,(FX(K),K=17,3
     13),NCARD2,ID2
      GO TO 105
  122 CONTINUE
C     IF NO ITEMS BEING ADDED OR MODIFIED, SKIP
      IF(ITEMAD.EQ.0) GO TO 222
    3 CONTINUE
      J=0
    4 CONTINUE
C READ # OF CARDS CONTAINING ITEM(STEM(J,K)) THAT IS BE ADDED TO TAPE
C     (INCLUDING NEW AND REVISED ITEMS.)
```

```
      J=J+1
      READ(5,110)(STEM(J,K),K=1,17),NQ(J),NPT(J)
      IF(J-1)63,63,59
   59 CONTINUE
C CHECK FOR CONSISTENT ITEM ID # (NQ(J))
      IF(NQ(J)-NQ(J-1))60,62,60
   60 CONTINUE
      WRITE(6,61)NQ(J),NQ(J-1)
   61 FORMAT(/////,1X,'****NOTE,NOTE,NOTE,NOTE****',/,6X,'MISMATCH OF ID
     1S WITHIN ITEM:',I4,' AND ',I4,/////)
      NWRONG=NWRONG+1
   62 CONTINUE
   63 CONTINUE
C CHECK FOR INVALID CHARACTER IN NPT(J). ALLOW C,E,*, OR BLANK
      IF(NPT(J).NW.IC.AND.NPT(J).NE.IBLK.AND.NPT(J).NE.IZ.AND.NPT(J).NE.
     1IE)GO TO 41
      GO TO 43
   41 CONTINUE
      WRITE(6,42) NQ(J), NPT(J), J
   42 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE NOTE',/,6X,'ITEM ',I4,'
     1HAS THIS INCORRECT ALPHABETIC ENDING:(',A1,') ON CARD',I3,/////)
      NWRONG=NQRONG+1
   43 CONTINUE
C IF E END OF TAPE IS SENSED
      IF(NPT(J).EQ.IE) GO TO 222
C END OF ITEM SENSED IF NPT(J) IS BLANK. THEN READS TWO PARAMETER CARDS
C ACCOMPANYING EACH ITEM.
      IF(NPT(J).NE.IBLK)  GO TO 4
      READ(5,103)(IX(I),I=1,22),(FX(K),K=1,16),NCARD1,ID1,(FX(K),K=17,33
     1),NCARD2,ID2
C     CHECK FOR MATCH OF ITEM ID#(NQ(J)) AND FIRST PARAMETER CARD ID #
     1 (ID1)IF(NQ(J)-ID1)56,58,56
   56 CONTINUE
      WRITE(6,57) NQ(J),ID1
   57 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE****',/,6X,'MISMATCH OF ID
     1S BETWEEN ITEM AND PARMETER CARD:',/,6X,'ITEM ID IS:',I4,/,6X,'PAR
     1AMETER ID IS:',I4,/////)
      NWRONG=NWRONG+1
      GO TO 3
   58 CONTINUE
C CHECK FOR ORDERING OF PARAMETER CARDS (NCARD1 AND NCARD2)
      IF(NCARD2-NCARD1)50,50,52
   50 CONTINUE
      WRITE(6,51) NCARD1,ID1,NCARD2,ID2
   51 FORMAT(/////,1X,'****NOTE NOTE NOTE NOTE ',/'5X,'PARAMETER CARDS A
     1RE OUT OF ORDER.',/,5X,'FIRST PARMETER CARD READS:CARD ',I1,2X'ITE
     2M # ',Ir,/,5X,'SECOND PARAMETER CARD READS: CARD ',I1,2X'ITEM #',I
     34,/////)
      NWRONG=NWRONG+1
      GO TO 3
   52 CONTINUE
```

```
C CHECK FOR MATCH OF PARAMTER CARDS ID #'S (ID1. AND ID2)
      IF(ID1-ID2)53,54,53
   53 CONTINUE
      WRITE(6,51) NCARD1,ID1,NCARD2,ID2
      NWRONG=NWRONG+1
      GO TO 3
   54 CONTINUE
      DO 801 L=1,J
      WRITE(1,110)(STEM(L,K),K=1,17),NQ(L),NPT(L)
      IF(L.EQ.1) GO TO 803
      GO TO 801
  803 CONTINUE
      WRITE(6,704) NQ(1)
  704 FORMAT(10X,'ITEM  ',I5,'   HAS BEEN ADDED AS A NEW OR MODIFIED ITE
     1M.')
  801 CONTINUE
      WRITE(1,103)(IX(I),I=1,22),(FX(K),K=1,16),NCARD1,ID1,(FX(K),K=17,3
     13),NCARD2,ID2
      GO TO 3
C     WRITE E IN COLUMN 80 FOR PURPOSE OF SENSING END ON NEW TAPE
  222 WRITE(1,110)(STEM(1,K),K=1,17),NQ(1),NPT(1)
      IF(NWRONG.GT.0) GO TO 501
      GO TO 503
  501 CONTINUE
      WRITE(6,500)NWRONG
  500 FORMAT(///,1X,'*** NOTE:',/,5X,'THIS ATTEMPT TO UPDATE DATA FILE H
     1AS NOT BEEN ACCURATELY DONE.',/,5X,'THERE ARE ',I4," MISTAKES MADE
     2.REGARD ABOVE MESSAGES TO CORRECT AND RUN THIS PROGRAM AGAIN.',//)
      GO TO 378
  503 CONTINUE
      WRITE(6,500)NWRONG
  502 FORMAT(///,1X,'*** NOTE:',/,5X,'TO BE SURE TAPE HAS BEEN PROPERLY
     1UPDATED RUN MEDSIRCH2 ASKING FOR ITEMS THAT WERE MODIFIED OR ADDED
     2.',//)
  378 CONTINUE
      STOP
      END
```

A P P E N D I X G

Hardware Requirements

H.1 WITHOUT MODIFICATIONS

Program	Amount of Core Needed for Executing	Blocking (2 buffers)	Total	Number of Tapes Needed	Number of Discs Needed
CHECK	4K		4K		
UTILITY	1K	15K	16K	1	
MEDSIRCH-1	10K	60K	70K	1	4
MEDSIRCH-2	10K	30K	40K	1	1
UPDATE	54K	30K	74K	2	

Minimal requirements for using implemented design is 74K core storage, 2 tapes, and 4 discs.

H.2 WITH MODIFICATIONS

Modifications possible:

1. use tapes in lieu of discs in MEDSIRCH-1 and MEDSIRCH-2.
2. do not block tapes or discs.
3. do not retrieve items at some or all lower hierarchical levels cf. pp. 57-58) in MEDSIRCH-1.

Note: If modifications (1) and (2) are used efficiency will be poorer.

Program	Amount of Core Needed for Executing	Blocking (2 buffers)	Total	Number of Tapes Needed	Number of Discs Needed
CHECK	4K		4K		
UTILITY	1K		1K	1	
MEDSIRCH-1	10K		10K	1-4	
MEDSIRCH-2	10K		10K	1	
UPDATE	54K		54K	2	

Minimal requirements for using modified implemented design is 54K core storage and 2 tapes.